VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Implementation of Non-Overlapping Domain Decomposition Techniques for FETI Methods

Doctor PhilosofiæThesis

Ing. Pavla Kabelíková

Ostrava, 2012

Study Programme:	Computer Science, Communication technology and Applied Mathematics (P1807)
Study Branch:	Computational and Applied Mathematics (1103V036)
Supervisor:	Doc. Mgr. Vít Vondrák, Ph.D.

Acknowledgments

Besides my supervisor Doc. Mgr. Vít Vondrák, Ph.D. and former head of the Department of Applied Mathematics Prof. RNDr. Zdeněk Dostál, DSc., I would like to thank to all of my colleagues from Department of Applied Mathematics who helped me during my Ph.D. study, especially to Mgr. Petr Kovář, Ph.D. who provided me valuable advices not only from the field of graph theory, to all of my colleagues from our former and current office, namely to Ing. Petr Beremlijski, Ph.D., Ing. Alexandros Markopoulos, Ph.D., and Ing. Oldřich Vlach Ph.D., who are always ready to discuss various topics to help anybody to understand the substance of problems, and finally to Ing. Michal Merta, who helped me during last days with finishing the final part of the work.

Last but not least, I would like to thank O. Univ. Prof. Dr. Phil. Wilfried Imrich and Ao. Univ. Prof. Mag. et Dr. rer. nat. Clemens Brand from the Department Mathematik und Informationstechnologie, Montanuniversität Leoben, who also supported me in my research on the field of graph theory.

Abstract

In this work, some improvements to algorithms for FETI (Finite Element Tearing and Interconnecting) methods together with theoretical background are presented. A smaller part of the text is devoted to provide an optimal domain decomposition for FETI method, which is essential for the further analysis.

The main part of this work is given to understand the meshes arising during discretization of numerical problems from graph theory point of view and to analyze these meshes as a graphs. This enables the use of some techniques of spectral graph theory directly applied to meshes of numerical problems.

The particular purpose of this analysis is to find certain nodes in meshes to reduce numerical instability in Cholesky-SVD method, especially when applied to solving discretized version of Neumann problem (stabilizing action of general inverse of semidefinite stiffness matrix). Partly guided by intuition about solid mechanics (vibration modes), it is natural to expect the so-called "fixing nodes" near the "center" of a graph.

Several candidates to these nodes are provided based on (spectral) graph techniques together with experimentally results. One of the assets of this work is the theoretical numerical analysis of one of these candidates confirming the idea that this candidate provides the best solution to the given problem, according to its definition.

Keywords

non-overlapping domain decomposition, FETI Methods, semi-definite matrices, generalised inverse, spectral graph theory, graph center, fixing nodes.

Abstrakt

V této práci jsou, společně s teoretickými podklady, prezentována vylepšení algoritmů pro FETI (Finite Element Tearing and Interconnecting) metody. Menší část je věnována zajištění vhodného rozložení oblastí pro FETI metody, což je nezbytný předpodklad pro další analýzu.

Větší část práce je věnována porozumění sítím vznikajícím z diskretizace numerických problémů z pohledu teorie grafů, což je důležité pro možnost analyzování těchto sítí jako grafů. Toto umožňuje použití některých technik z oblasti spektrální teorie grafů přímo na sítě numerických problémů.

Konkrétní účel této analýzy je nalezení jistých vrcholů v sítích pro redukci numerické nestability v Cholesky-SVD metodě, zvláště když je aplikována na řešení diskretizované verze Neumannovy úlohy (stabilizace akce zobecněné inverze semidefinitní matice tuhosti). Částečně vedeni intuicí z oblasti mechaniky (vibrační módy), přirozeně očekáváme tyto, tak zvané, "fixující vrcholy" poblíž "centra" grafu.

Společné s experimentálními výsledky zde uvádím několik možností, jak nalézt kandidáty na fixující vrcholy, založených na (spektrálních) grafových technikách. Jeden z přínosů práce je teoretická numerická analýza jednoho z těchto kandidátů, která potvrzuje myšlenku, že tento kandidát poskytuje nejlepší řešení daného problému ve smyslu definice.

Klíčová slova

nepřekrývající se dělení oblastí, FETI metody, semi-definitní matice, zobecněná inverze, spektrální grafová teorie, grafová centra, fixující uzly.

Contents

List of Figures

List of Tables

Introduction

Finite Element Tearing and Interconnecting (FETI) methods are a powerful tool for solving various practical problems, e.g., solid mechanics, shape optimization, contact problems, fluid dynamics, etc. For parallel implementation of the FETI methods, some general tool is necessary to decompose the given problem into prescribed number of non-overlapping subdomains. To obtain a successful solution of the resulting problem, it is essential that all the subdomains are simply connected and the decomposition is of high quality.

While solving a system of consistent linear equations with symmetric positive semidefinite (SPS) matrix arising in the stress analysis of a "floating" static structure whose essential boundary conditions are not sufficient to prevent its rigid body motions, it arises the need of computation of a generalized inverse matrix [25, 26, 54]. This system can be solved by standard direct methods for the solution of systems with positive definite matrices, such as the Cholesky decomposition, adapted to the solution of systems with only positive semidefinite matrix. The only modification comprises setting to zero these columns, which correspond to zero pivots. However, in agreement with the theoretical results of Pan [53], it turns out that it is very difficult to recognize the positions of such pivots in the presence of rounding errors when the nonsingular part of A is ill-conditioned. Due to these rounding errors, the main difficulty in implementation of the FETI method is an effective elimination of the displacements, i.e., evaluation of the action of generalized inverse of the SPS stiffness matrices of "floating" subdomains. To alleviate this problem, Farhat and Géradin [25] proposed to combine the Cholesky decomposition with the SVD decomposition of a relatively small matrix. The method was developed further by Papadrakakis and Fragakis [54]. An improved modification of combination of the Cholesky decomposition with the SVD decomposition is proposed by Brzobohatý, Dostál, Kozubek, Kovář, and Markopoulos in [9]. This modification based on the active choice of the SVD part uses fixing nodes strategy to make the system as stiff as possible and has been implemented in the Total FETI solver. This solver uses the La-

grange multipliers not only to glue the subdomains along auxiliary interfaces, but also to implementation of the essential boundary conditions; first considered by Felipa, Park, Justino, and Gumaste [26]; then by Dostál, Horák, and Kučera [18]. The main advantage of this approach is that it makes all the subdomains floating, so that the null spaces of the stiffness matrices are *a priori* known.

The asset of this thesis is to present several improved algorithms and new techniques arising while solving various problems of numerical mathematics, computational mechanics, etc. In particular, a new insight into advanced numerical mathematics from the point of view of (spectral) graph theory is presented. The whole work is divided into five chapters.

Chapter 1 establishes a given numerical problem from the point of view of FETI methods and introduces some of basic domain decomposition algorithms. A motivation example from contact mechanics is described in Section 1.1. This model example could be considered further in the thesis when some "problem" is mentioned. In Section 1.2, the domain decomposition is mentioned as an essential part of parallel computation of FETI methods. A comprehensive overview about mainstreams in partitioning techniques is presented as geometric partitioning, hypergraph and graph partitioning, and several best known software and packages based on these techniques. Also, the optimal decomposition for needs of FETI methods is defined. Unfortunately, none of these software packages is able to provide such a decomposition. Hence, post-processing on output of METIS software [47] is proposed to compute the optimal decomposition.

Chapter 2 deals with spectral theory. At first for the continuous case, later for the discrete case from graph theory point of view. Because there is a parallel between continuous and discrete case, Section 2.1 brings an overview about this topic also for the continuous case. Using the FETI algorithm on certain type of problems, it turns out that some parts of the solution can be solved by means of graph theory. Thus, in Section 2.2, basic notation from the graph theory field is establish which is used further in this work. In Section 2.3, there is summarized an overview about spectral graph theory. There are some results on the frontier between graph theory and linear algebra that could be suitable for understanding of spectral properties of graphs well. Especially, Fiedler based contributions to graph theory, interlacing properties and notes about spectrum of Cartesian product are widely used in this work. The spectrum of certain types of graphs such as paths or Cartesian products of two paths is analyzed in Section 2.4.

Author's results are described in Chapter 3. This chapter is focused on the strategy to find generalized inverse of singular matrix based on, so-called,

fixing nodes. Section 3.1 collects an overview about the generalized inverse and its role in FETI algorithms. In this section, the problem of finding fixing nodes is established. In Section 3.2, author's new definitions of fixing node(s) are introduced. Several candidates to approximate fixing node are discussed in Section 3.3 together with experimentally proven results. In this section, a new approach how to use (spectral) graph theory techniques arises in an effective solution of linear systems of equations with SPS matrices.

One of the valuable parts of this work is contained in Chapter 4, where the analysis of one of the discussed candidates is made to establish that this candidate provides the best choice of one–fixing node according to definition. As the proof is technically extensive, the best choice of one–fixing node is reformulated into another definition in Section 4.1. Several preliminaries for following consideration are established in Section 4.2. In Section 4.3, the easiest example is computed and the same approach is finally extended to n variables (in one–dimension) in Section 4.4. In Section 4.5, necessary steps are presented to extend these considerations into two–dimensions, which is similar approach but with more complicated expressions.

The last chapter is devoted to experiments. In Section 5.1, two approaches for finding fixing nodes are compared. The one used by T. Brzobohatý, Z. Dostál, T. Kozubek, P. Kovář and A. Markopoulos in [9] and the new one introduced in this thesis. In Section 5.2, solutions of two contact problems using FETI algorithm with the fixing node strategy are shown. The last section, Section 5.3, shows that even the large-scale problems can be effectively solved using the fixing node strategy.

Chapter 1

FETI Methods

1.1 Motivation: FETI Methods

At the beginning of this section, a model example is established. This example can be discretized and solved numerically by some appropriate method. As it was mentioned in introduction, we usually use one of FETI method. A brief overview of FETI methods is presented in Subsection 1.1.3.

1.1.1 Model Contact Problem

Let us consider a two-dimensional frictionless contact problem depicted in Figure 1.1, adopted from [32]. Two bodies are denoted by $\Omega^i \in \mathbb{R}^2$, $i = 1, 2$, and their boundaries Γ^i are decomposed into disjoint parts Γ^i_u for Dirichlet boundary condition, Γ^i_p for Neumann boundary condition and Γ^i_c for contact non-penetration condition, so that $\partial\Omega^i = \Gamma^i_u \cup \Gamma^i_p \cup \Gamma^i_c$.

On Γ^i_u, the zero displacements are prescribed, on Γ^i_p, the surface tractions of density p^i act, and on Γ^i_c, two contact conditions are considered: non-penetration of the bodies and the transmission of the contact stress. Each body Ω^i is subject to volume forces of density f^i.

Let us seek the displacement fields u^i in Ω^i satisfying the equilibrium equations and the Dirichlet and Neumann boundary conditions:

$$\left.\begin{array}{rcccl} -\mathrm{div}\sigma^i & = & f^i & \text{in} & \Omega^i, \\ u^i & = & 0 & \text{on} & \Gamma^i_u, \\ \sigma^i n^k & = & p^i & \text{on} & \Gamma^i_p, \end{array}\right\} i = 1, 2, \tag{1.1}$$

where $\sigma^i := \sigma(u^i)$ is the stress tensor in Ω^i and n^i stands for the unit outward normal vector to $\partial\Omega^i$, $i = 1, 2$. Stress tensors are related to the linearized strain tensors by the Hooke's law for linear isotropic material, see [32].

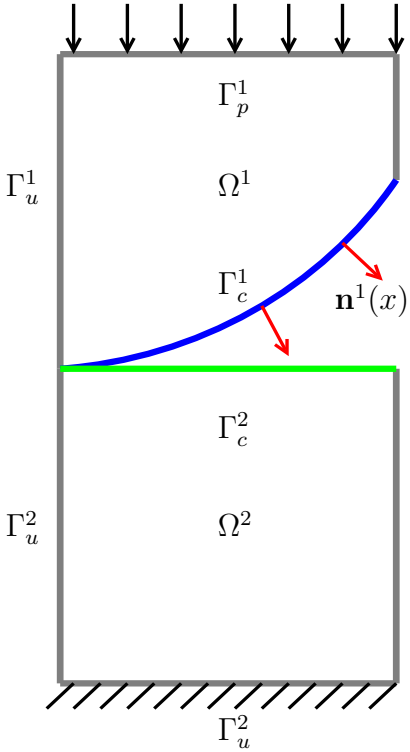


Figure 1.1: Contact problem

To formulate the contact conditions, an one-to-one transfer mapping $\chi := \Gamma_c^1 \rightarrow \Gamma_c^2$ has to be predefined. The initial distance between the contact surfaces at $x \in \Gamma_c^1$ is computed as $d(x) := \|\chi(x) - x\|$ and the critical direction as $\nu(x) := (\chi(x) - x)/d(x)$ if $d(x) \neq 0$, or $\nu(x) := n^1(x)$ if $d(x) = 0$. The $\| \, . \, \|$ stands for the Euclidean norm of a vector. The non-penetration conditions can be written as follows:

$$\left.\begin{array}{rcl} u_\nu - d & \leq & 0, \\ \sigma_\nu & \leq & 0, \\ \sigma^\nu(u_\nu - d) & = & 0, \end{array}\right\} \text{ on } \Gamma_c^1, \tag{1.2}$$

where $u_\nu(x) := (u^1(x) - u^2(\chi(x)))^T \nu(x)$ is the relative contact displacement and $\sigma_\nu(x) := \nu(x)^T \sigma^1(x) \nu(x)$ is the contact stress at $x \in \Gamma_c^1$ and both in the direction of $\nu(x)$.

The transmission of the contact stress can be written as

$$\sigma^1 = \sigma^2 \text{ on } \Gamma_c^1. \tag{1.3}$$

The weak formulation of the contact problem, see [33], can be written in the following form:

$$\text{find } u \text{ such that } J(u) = \min_{v \in \mathbb{K}} J(v), \tag{1.4}$$

where

$$J(v) = \frac{1}{2} a(v, v) - b(v) \tag{1.5}$$

and

$$\mathbb{K} = \{v \in \mathbb{V} | v_\nu - d \leq 0 \text{ on } \Gamma_c^1\},$$
$$\mathbb{V} = \{v := (v_1, v_2) \in (H^1(\Omega^1))^2 \times (H^1(\Omega^2))^2 | v^k = 0 \text{ on } \Gamma_u^k,\ k = 1, 2\}.$$

In (1.5), $a(v, v)$ denotes a bilinear form representing the inner energy of given problem and $b(v)$ denotes the linear form representing the work of forces f^i, $i = 1, 2$.

1.1.2 Discretization of Contact Problem

For numerical solution, the problem represented by its weak formulation (1.4) is discretized. One of appropriate methods is Finite Element Method (FEM).

After the discretization, the energy functional $J(v)$ is of the form

$$J(v) = \frac{1}{2} v^T K v - v^T f, \tag{1.6}$$

where K is the stiffness matrix, f is the right-hand side vector, $v \in V_h$, where V_h is a space of piecewise polynomial functions. Solution of this problem consists in solving appropriate system of linear equation.

1.1.3 FETI Algorithms

The Finite Element Tearing and Interconnecting (FETI) method first presented by Farhat and Roux in [24] turned out to be one of the most appropriate algorithm for parallel solution of problems described by elliptic partial differential equations (PDEs).

Implementation of the standard FETI methods (FETI-1, FETI-2) requires identification of the kernels of the stiffness matrices of the subdomains. In opposite, in the FETI-DP method, all stiffness matrices are invertible.

Problem (1.8) is solved in dual variables λ instead in the primal variables u to decrease order of the problem. Saddle point formulation of some problem seems in all FETI methods similar:

$$L(u, \lambda) = \frac{1}{2} u^T K u - f^T u + \lambda^T B u, \tag{1.7}$$

where u is the vector of primal unknowns, λ is the vector of the Lagrange multipliers (dual unknown), K is the stiffness matrix, f is the initial force and B is the matrix of boundary conditions.

The problem (1.7) is equivalent to the saddle point problem

$$\text{find } (\overline{u}, \overline{\lambda}) \text{ so that } L(\overline{u}, \overline{\lambda}) = \sup_{\lambda \geq 0} \inf_{u} L(u, \lambda). \tag{1.8}$$

Exact solution of the problem (1.8) differs depending on the variant of the FETI method. All of these variants are widely described, for example, in [36]. Here, I present only brief description of the best known variants.

FETI-1

The standard FETI method (FETI-1) first published by Farhat and Roux in [24] is based on the decomposition of the spatial domain into non-overlapping subdomains that are "glued" by Lagrange multipliers. Efficiency of this method can be further improved by implementing special projectors and preconditioners.

The stiffness matrices are symmetric positive definite (SPD) or symmetric positive semidefinite (SPS) matrices. The SPS matrix is singular, thus the corresponding inverse matrix does not exists and some generalized inverse matrix has to be found, which is the bottleneck of entire FETI-1 method. However, some algorithms to find a good generalized inverse are known and one new approach to find the generalized inverse is presented in this text, as well as author's own contribution to this topic.

FETI-2

The FETI-2 method (first considered by Farhat *et al.*) differs from FETI-1 in projecting the Lagrange multipliers at the cross-points in each iteration to enforce continuity of the primal solutions. This produce faster convergence for several problem classes.

FETI-DP

FETI-DP (Dual-Primal) introduced again by Farhat *et al.* in [23] has similar effect as FETI-2. The continuity of the primal solution at cross-points (*corners*) is implemented directly into the formulation of the primal problem. The continuity of the primal variables across the rest of the subdomain interfaces is again enforced by the Lagrange multipliers.

There are some possibilities how to define corners, but all of them preserve the invertibility of the stiffness matrices of subdomains. The bottleneck consisting in computation of generalized inverse vanishes.

Total-FETI

The last variant of the FETI method, FETI-1 respectively, is the TotalFETI (sometimes called TFETI-1) method. The main idea of this method consists in using the Lagrange multipliers not only for gluing of the subdomains along the auxiliary interfaces but also for implementation of the (Dirichlet) boundary conditions. This modification causes that all stiffness matrices corresponding to subdomains are "floating" and the method is thanks to this modification easier to implement. This method is described for example in [18].

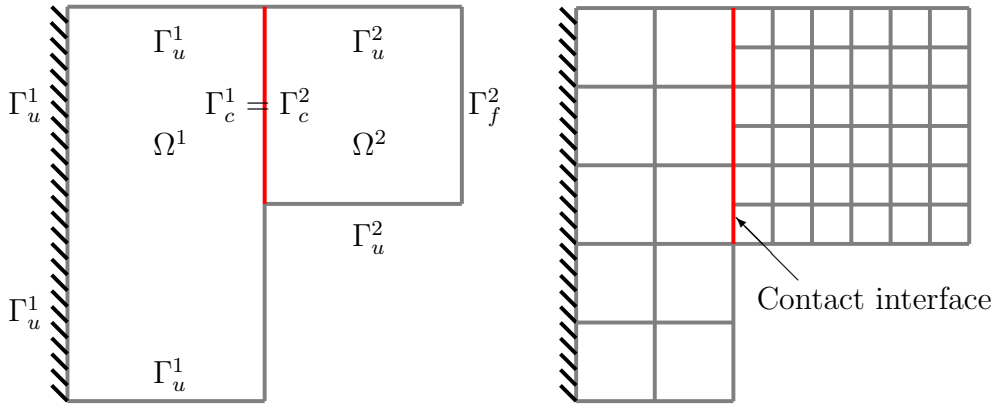
In original FETI-1 method, the kernels of stiffness matrices were not *a priori* known, which has caused problems in computation of the generalized inverse. In Total-FETI, all stiffness matrices of subdomains have the same kernel, therefore they can be processed in the same way.

1.2 Techniques of Domain Decomposition

It is well known, that in FETI methods a problem of domain decomposition arises. The idea of FETI methods can be found in [24]. The FETI algorithms work with non-overlapping domain decomposition methods (sometimes called as sub-structuring methods). The domain decomposition is the essential part of the FETI algorithms. Requirements to the optimal domain decomposition are presented in Subsection 1.2.1. There are a lot of software packages developed to provide the domain decomposition, several of them are mentioned in the Subsection 1.2.2 of this section. In the Subsection 1.2.3, the several improvements how to obtain "better" decomposition are described.

1.2.1 Non-Overlapping Domain Decomposition

The methodology is adapted to the solution of frictionless multi-body contact problems. In this case, each solid body is usually considered to be a separate domain as we can see in Figure 1.2(a). The notation is similar to that used in Subsection 1.1.1.



(a) Basic problem (b) Discretized problem

Figure 1.2: Basic notation of the contact problem

Applying finite element method to the problem from Figure 1.2(a), discretization depicted in Figure 1.2(b) is obtained. In Figure 1.2(b), we can see applied Dirichlet boundary conditions, displacements u_j associated with the nodes and the contact interface.

In case of large problems the need of parallelism arises. However, the decomposition from Figure 1.2 is not sufficient for purposes of parallel computing. The basic idea is to decompose given problem such that one domain

will be assigned to one processor. In practical examples, spatial discretization meshes are typically very complex for use of some intuitive partitioning technique and therefore more powerful partitioning tools are needed, e.g. Chaco [38], Metis [47], Jostle [64], Scotch [58].

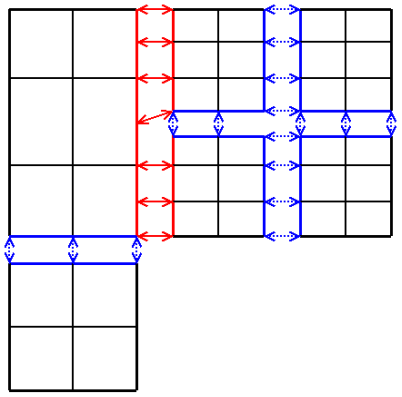


Figure 1.3: Optimal decomposition

Let us apply the non-overlapping domain decomposition to the problem depicted in Figure 1.2(a) that consists of two solid bodies. In Figure 1.3, the problem is decomposed into six subdomains. The original contact interface is considered as a part of the interface between the solid bodies and it is depicted by the red solid arrows. The new contact interface is the part of the interface between all subdomains, i.e. the the blue dashed arrows together with the red solid arrows. The continuity of the solution along all subdomains is guaranteed by means of Lagrange multipliers.

For the optimal decomposition several conditions should be preserved:

Definition 1.2.1 Optimal decomposition.

1. Each subdomain has to be connected.
   This condition guarantees the existence of TFETI solution for well posed problems.
2. The fill-ins in each subdomain should be approximately equal or, at least, the number of elements in each subdomain should be approximately equal.
3. The number of edges on interfaces among all subdomains should be minimized.
4. The shape of each subdomain should be as compact as possible.
   Points 2–4 guarantee a good load balancing.

1.2.2 Domain Decomposition Software Tools for FETI Methods

Non-overlapping domain decomposition is obtained by partitioning of a mesh. Usually, the mesh is converted to a graph, obviously non-oriented, unweighted. Depending on strategy, the graph is created either directly from the mesh, i.e. nodes of the original mesh are taken as vertices of the new graph, or the graph is created as, so-called, dual graph. In case of the dual graph, the vertices of the new graph are created from the elements of the original mesh, and the new vertices are connected by a new edge if the original elements were adjacent. For domain decomposition, the dual graph strategy is used, because the decomposition on element level is required. The procedure of creating such a graph is described, for example, in [47].

There are various approaches to partition the mesh. Here, I would like to present some general overview of these approaches adopted from [68].

Geometric (Coordinate-based) Partitioning

Geometric partitioners divide data into partitions based on the physical coordinates of the data. Objects assigned to a single partition tend to be physically close to each other in space. Such partitioners are very useful for applications that do not have explicit connectivity information or for which geometric locality is important because they are mesh independent. They might be used in adaptive finite element methods because, in general, they execute very quickly and yield quite good partition quality. The advantage of geometric partitioners is that they are the easiest non-trivial partitioners and they require only several passing through the list of nodes (say 2-3) to complete the partitioning.

The list of geometric partitioners follows.

1. **Recursive Coordinate Bisection (RCB)**
   RCB was first proposed as a static load-balancing algorithm by Berger and Bokhari in [5], but is attractive as a dynamic load-balancing algorithm because it implicitly produces incremental partitions. In RCB, the computational domain is first divided into two regions by a cutting plane orthogonal to one of the coordinate axes so that half the work load is in each of the sub-regions. The splitting direction is determined by computing in which coordinate direction the set of objects is most elongated, based upon the geometric locations of the objects. The sub-regions are then further divided by recursive application of the same splitting algorithm until the number of sub-regions equals the number of processors. Although this algorithm was first devised to cut into a

number of sets that is a power of two, the set sizes in a particular cut need not be equal. By adjusting the partition sizes appropriately, any number of equally-sized sets can be created. If the parallel machine has processors with different speeds, sets with non-uniform sizes can also be easily generated.

2. **Recursive Inertial Bisection (RIB)**
   RIB was proposed as a load-balancing algorithm by Williams in [67] and later studied by Taylor and Nour-Omid in [62], but its origin is unclear. RIB, similarly to RCB, divides the domain based on the location of the objects being partitioned by use of cutting planes. In RIB, the computational domain is first divided into two regions by a cutting plane orthogonal to the longest direction of the domain so that half the work load is in each of the sub-regions. The recursive procedure how to obtain more regions is similar to that used in RCB.

3. **Tree-based methods**
   All tree-based methods use a tree that represents the process of partitioning, for example Octree partitioners, Hilbert Space Filling Curve (HSFC), Refinement Tree Partitioning (REFTREE, [51]), etc.

Implementations of these geometric partitioners could be found in several software, for example, in Zoltan [68].

For purposes of domain decomposition, we do not use these types of partitioners from several reasons. First, sometimes the coordinates are not available with the mesh. Second, for irregular and non-homogeneous meshes, as well as for meshes with refinement on the contact interface, these techniques sometimes do not provide optimal decomposition in sense of Definition 1.2.1.

Hypergraph Partitioning

Hypergraph partitioning is a useful partitioning and load balancing method when connectivity data is available. A hypergraph consists of vertices and hyperedges. A hyperedge connects one or more vertices. A graph can be cast as a hypergraph in two ways: either every pair of neighbouring vertices form a hyperedge, or a vertex and all its neighbours form a hyperedge. The hypergraph model is well suited to parallel computing, where vertices correspond to data objects and hyperedges represent the communication requirements. The basic partitioning problem is to partition the vertices into k approximately equal sets such that the number of cut hyperedges is minimized. Most partitioners allows a more general model, where both vertices and hyperedges can be assigned weights. It has been shown, that the hypergraph

model gives a more accurate representation of communication cost (volume) than the graph model. In particular, for sparse matrix-vector multiplication, the hypergraph model exactly represents communication volume. Sparse matrices can be partitioned either along rows or columns; in the row-net model the columns are vertices and each row corresponds to an hyperedge, while in the column-net model the roles of vertices and hyperedges are reversed.

Implementation of hypergraph partitioning could be found for example in hMETIS [47], ZOLTAN [68], or PaToH [10] packages.

Graph Partitioning

Partitioning techniques without using coordinates are more sophisticated in sense of graph theory. Sometimes, the pure graph theory as a tool for computation of partitioning is used, sometimes, the connectivity of the graph is used for another computations techniques.

Graph partitioning has turned out to be quite useful in scientific computing. In contrast to hypergraph partitioning, graph partitioning does not contain hyperedges that connect two or more vertices (only two vertices could be connected) and usually, it takes less computational time.

The main idea of graph partitioning is to represent the computational application as a weighted or unweighted graph. Edges in the graph usually correspond to communication costs. In graph partitioning, the problem is to find a partitioning of the graph (that is, each vertex is assigned to one out of k possible sets called partitions) that minimizes the cut size (weight) subject to the partitions having approximately equal size (weight). This problem is NP-complete so no efficient exact algorithm is known, but heuristics work well in practice.

There are two main approaches based on structure of a mesh.

1. Spectral graph partitioning,

2. multilevel graph partitioning.

To get some informations about spectral graph partitioning, see, for example, [4], [40], [59]. Multilevel approaches have turned to be faster and better in sense of optimal partitioning, therefore, we aim on them.

The best known representative of multilevel approach is the METIS software developed by G. Karypis and V. Kumar at the University of Minnesota [47]. The parallel version of the METIS is called ParMETIS. METIS or ParMETIS are not methods but collections of methods that provide tools for mesh and graph partitioning.

Chaco by B. Hendrickson and R. Leland from Sandia National Laboratories [38] is also based on the graph partitioning. As usual, Chaco is a package that implements wider functionality. Chaco allows recursive application of several methods for finding small edge separators in weighted graphs. These methods include inertial, spectral, Kernighan-Lin, and multilevel methods in addition to several simpler strategies.

The next representative is the Jostle. Jostle is a library for graph (mesh) partitioning and load balancing developed by Chris Walshaw at the University of Greenwich [64]. The main algorithm used in Jostle is based on multilevel graph partitioning, and a diffusion-type method is available for repartitioning. Therefore, the Jostle library is very similar to ParMETIS.

Another refereed software dealing with partitioning are, for example, Party [60], Ralpar [30], Scotch [58], Zoltan [68].

Approach presented in this section can be similarly applied to the solution of other problems with stiffness matrix of non-trivial kernel, as well as it can be extended to the three-dimensional problem accordingly.

1.2.3 Connected Domain Decomposition

The METIS software is very fast and it usually provides very good solution in the sense of the optimal decomposition. However, it sometimes does not provide connected[1] subdomains as one can see in Figure 1.5(a). Thus, some post-processing on partitioning obtained from METIS has to be done.

The basic idea of implemented post-processing is described bellow.

Algorithm (post-processing).

1. For each subdomain: test, if it is connected.
   1.1. If yes, continue,
   1.2. else
2. Find all its parts using Breath First Search algorithm.
3. Leave the part with the maximum number of nodes.
4. The remaining parts join to their neighbour subdomains.
   To preserve the compactness, I choose the one from the neighbour subdomains that shares with the chosen part the maximum number of nodes.

The Breath First Search (BFS) algorithm is used also for testing the connectivity of domains. The description of this algorithm can be found, for example, in [41].

[1] Here, the adjectives "connected" or "disconnected" before the word subdomain means that the graph corresponding to the mesh is connected or disconnected, respectively.

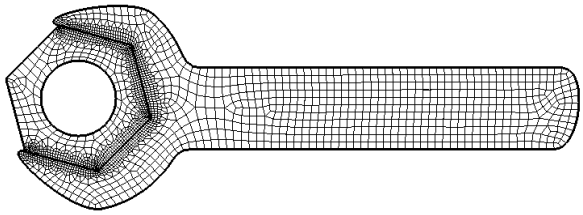
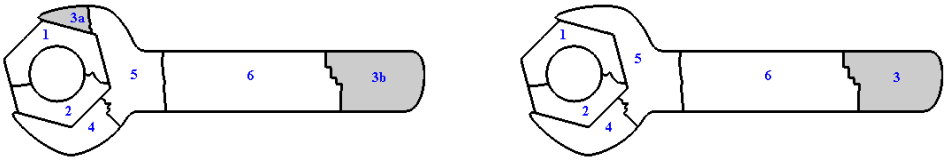


Figure 1.4: Testing mesh



(a) Disconnected decomposition (b) Connected decomposition

Figure 1.5: Decomposition of given problem

Let us have a look on the mesh of testing problem depicted in Figure 1.4. The problem consists of two solid bodies: a screw female and a spanner, where the screw female is fixed and the spanner is floating. Assume that one can decompose this mesh into six subdomains using METIS partitioning tool. The original output from this partitioning tool is depicted in Figure 1.5(a). One can see that the subdomain 3 is disconnected, which is in contradiction to optimal decomposition. The modified decomposition after author's post-processing is depicted in Figure 1.5(b).

Chapter 2

Spectral Theory & Graphs

2.1 Notes on Spectrum: Eigenvalue Problem

The purpose of this section is to give an introduction and a background for the following Section 2.3 which is mostly based on the book of Biyikoglu, Leydold, and Stadler: *Laplacian Eigenvectors of Graphs* [6]. They have (first) given wide overview of known properties of eigenvectors of graphs, completed with their own investigations and remarks.

Usually, eigenvalues of graphs have been defined as eigenvalues of the corresponding adjacency matrix. They have been used in investigation of various graph properties, for example for characterizing classes of graphs, for obtaining bounds on properties such as the diameter, girth, chromatic number, connectivity, etc.

Later, the interest has been shifted from the adjacency spectrum to the spectrum of closely related graph Laplacian, but, before the book of Biyikoglu, Leydold, and Stadler, no one has paid as much attention to the spectrum of graph Laplacian and to the eigenvectors of graph Laplacian especially.

However, their investigation requires some introduction and comments. For better insight into this theory, let us start with classical definition of the (linear algebra) eigenvalue problem.

Definition 2.1.1 (Eigenvalue problem)
Given a matrix $A \in \mathbb{R}^{n\times n}$, a non-zero vector $v \in \mathbb{R}^n$ is defined to be the eigenvector *of the matrix A if it satisfies the eigenvalue equation*

$$A \cdot v = \lambda \cdot v, \tag{2.1}$$

for some scalar $\lambda \in \mathbb{R}$. In this situation, the scalar λ is called an eigenvalue

of A corresponding to the eigenvector v of the dimension n [1].

An eigenspace *is the set of all eigenvectors with the same eigenvalue, together with the zero vector.*

Notice, that the classical linear algebra definition deals in the finite-dimensional space with finite matrices and vectors (*discrete case*).

Another approach how to look at the eigenvalue problem is the continuous formulation based on infinite-dimensional spaces of function. We can consider similar (eigenvalue) equation written in the form

$$\mathcal{A} \cdot f = \lambda \cdot f, \tag{2.2}$$

where a non-zero function $f : V \to \mathbb{R}$ is an *eigenfunction* of a linear operator $\mathcal{A}$, defined on some function space. Function f returns from the operator $\mathcal{A}$ exactly as is, except for a multiplicative scaling factor λ, the corresponding *eigenvalue*.

If $\mathcal{A}$ is, for example, the differential operator, the solution of the differential eigenvalue problem could depend on some boundary conditions required of f. Obviously, there are only certain eigenvalues $\lambda = \lambda_n$ $(n = 1, 2, 3, \ldots)$ that admit a corresponding solution for $f = f_n$ $(n = 1, 2, 3, \ldots)$ (with each f_n belonging to the eigenvalue λ_n) when combined with the boundary conditions.

To show some model example, let us present the example from [65] (Eigenvalue, eigenvector and eigenspace): *"A common example of maps on infinite dimensional spaces are the action of differential operators on function spaces. As an example, on the space of infinitely differentiable functions, the process of differentiation defines a linear operator. The eigenvalue equation for linear differential operators is then a set of one or more differential equations. The eigenvectors are commonly called eigenfunctions. The most simple case is the eigenvalue equation for differentiation of a real valued function by a single real variable. We seek a function (equivalent to an infinite-dimensional vector) which, when differentiated, yields a constant times the original function. In this case, the eigenvalue equation becomes the linear differential equation*

$$\frac{d}{dx}(f)(x) = \lambda f(x). \tag{2.3}$$

Here λ is the eigenvalue associated with the function, $f(x)$. This eigenvalue equation has a solution for any value of λ. If λ is zero, the solution is

[1] In following sections, the dimension n will be equal to number of vertices of given graph, i.e. $n = |V|$.

$$f(x) = a, \tag{2.4}$$

where a is any constant; if λ is non-zero, the solution is the exponential function

$$f(x) = ae^{\lambda x} \ .” \tag{2.5}$$

2.1.1 Wave equation – Vibrating Problem

We give an introduction to the, so-called, *vibrating problem*. For this purpose, let us to introduce the wave equation which is a linear partial differential equation, that can be described by a linear differential operator – the special case of the eigenvalue problem described by Equation 2.2.

The *wave equation* is an example of a hyperbolic equation. The wave equation refers to a scalar function $u = (x_1, x_2, ..., x_n, t)$ that satisfies:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \triangle \overline{u}, \tag{2.6}$$

where $\triangle$ is the Laplace operator (Laplacian)[2] applied to $\overline{u}$, $\overline{u} = (x_1, x_2, ..., x_n)$, and c is a fixed constant equal to the propagation speed of the wave.

Special boundary value problems (BVP's) corresponding to equation (2.6) could be solved, for example, by the method known as *separation of variables*. Using this method, we can obtain separate solutions in $x = (x_1, \ldots, x_n)$ and t.

The wave equation has various applications. Solving the wave equation with boundaries in one space dimension (i.e. $x = (x_1) \in \mathbb{R}$), the *vibrating string* is obtained.

Vibrating string with ”fixed boundaries”

Let us have a look on the solution of the wave equation (2.6) in one dimensional space, so-called vibrating string problem. Wave equation (2.6) in one dimension can be rewritten as

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}. \tag{2.7}$$

[2] The Laplacian is defined as the divergence ($\nabla \cdot$) of the gradient (∇f)

$$\triangle f = \nabla^2 f = div\ grad\ f, \text{ i.e. } \triangle f = \nabla \cdot \nabla f$$

and it is a linear operator taking functions into functions.

Let us consider the case $c = 1$ and assign $u_{tt} = \frac{\partial^2 u}{\partial t^2}$ and $u_{xx} = \frac{\partial^2 u}{\partial x^2}$. Together with boundary conditions (2.9) and conditions to initial displacement $(u(x, 0))$ and velocity $(u_t(x, 0))$ (2.10), the complete formulation of BVP for the one dimensional space wave equation is obtained.

$$\begin{aligned} u_{tt}(x,t) &= u_{xx}(x,t), \quad x \in (0,L) \quad t \in (0,\infty), && (2.8)\\ u(0,t) &= 0, \quad u(L,t) = 0 \quad t \in (0,\infty), && (2.9)\\ u(x,0) &= 0, \quad u_t(x,0) = \psi(x) \quad x \in (0,L). && (2.10) \end{aligned}$$

Equations (2.8)–(2.10) describe a flexible string that is stretched between two points $x = 0$ and $x = L$ and satisfies the wave equation for $t > 0$ and $0 < x < L$. On the boundary points, u should satisfy the, so-called, Dirichlet boundary conditions, that correspond to fixed ends of the string. We can use the method of separation of variables and look for solutions of this problem in the special form

$$u(x,t) = X(x) \cdot T(t). \tag{2.11}$$

Substituting $u(x, t)$ into (2.8) and dividing both sides by $X(x) \cdot T(t)$, we get

$$\frac{T''(t)}{T(t)} = \frac{X''(x)}{X(x)} \quad \big(= -\lambda\big). \tag{2.12}$$

The left and right side of Equation (2.12) do not depend on the x and t variable, respectively, so they are equal to some constant $-\lambda$. We can split the equation into two independent equations and solve them separately for $T(t)$ and $X(x)$.

$$\begin{aligned} T'' + \lambda T &= 0, && (2.13)\\ X'' + \lambda X &= 0, && (2.14) \end{aligned}$$

Solving equations (2.14) with respect to boundary conditions (2.9)[3] we obtain the sequence of eigenfuctions $\{X_n(x)\}$ with corresponding sequence of eigenvalues $\{\lambda_n\}$ (of operator $\triangle\overline{u}$ in Equation 2.6),

$$X_n(x) = \sin\left(\frac{n\pi x}{L}\right), \quad \lambda_n = \left(\frac{n\pi x}{L}\right)^2, \quad n = 1, 2, 3, \dots \ . \tag{2.15}$$

Analogously, solutions (2.13) can be written in the form

$$T_n(t) = a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right), \quad n = 1, 2, 3, \dots \ . \tag{2.16}$$

[3]Remark, that we do not care about the trivial zero solution which also satisfies our equation. Thus, all relevant constants are forced to be non-zero.

Consequently, every linear combination $\sum_{k=1}^{\infty} X_k(x) \cdot T_k(t)$ represents solution of the problem (2.8) satisfying conditions (2.9). Additional assumptions on $\psi(x)$ from condition (2.10) can be done to ensure the possibility of introducing solution of the whole problem (2.8)– (2.10). The whole solution has the form

$$u(x,t) = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{L}\right) \left[a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right)\right], \quad n = 1, 2, 3, \dots \ . \tag{2.17}$$

By application of the condition to the initial displacement $u(x,0) = 0$ (2.10) we obtain coefficients $a_n = 0$ immediately. Coefficients b_n could be specified via Fourier sines series using the condition to the initial velocity $u_t(x,0) = \psi(x)$ (2.10):

$$b_n = \frac{2}{n\pi} \int_0^{\pi} \psi(x) \sin\left(\frac{n\pi x}{L}\right), \quad n = 1, 2, 3, \dots \ . \tag{2.18}$$

Finally, solution $u(x,t)$ depends on the selection of the initial velocity $\psi(x)$ and it can be computed in the form

$$u(x,t) = \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{n\pi t}{L}\right), \quad n = 1, 2, 3, \dots \ . \tag{2.19}$$

For the fixed choice of $n = k$ we obtain the k-th component of the sum (with b_k computed by (2.18)):

$$u_k(x,t) = b_k \sin\left(\frac{k\pi x}{L}\right) \sin\left(\frac{k\pi t}{L}\right). \tag{2.20}$$

Remark 2.1.2 (Laplace operator in wave equation)
Equation (2.14) can be written in the form $\triangle X(x) = \lambda \cdot X(x)$, (for this case operator $\triangle = -d^2/dx^2$) which is the eigenvalue problem corresponding to the wave equation.

The $\lambda_n = \left(\frac{n\pi x}{L}\right)^2$ are the eigenvalues of the Laplace operator together with appropriate homogeneous Dirichlet conditions on boundary (2.9) and functions $X_n(x) = \sin\left(\frac{n\pi x}{L}\right)$ are theirs corresponding eigenfunctions.

Vibrating string with "free boundaries"

Another situation occurs when one would like to analyze the vibrating string with "free" boundaries, i.e. when Neumann boundary conditions are used instead the Dirichlet conditions. Again, we consider the wave equation in

the form (2.7) with $c = 1$ on the same intervals $t > 0$ and $0 < x < L$ but with different boundary conditions.

$$\begin{aligned} u_{tt}(x,t) &= u_{xx}(x,t), \quad x \in (0,L) \quad t \in (0,\infty), && (2.21)\\ u_x(0,t) &= 0, \quad u_x(L,t) = 0 \quad t \in (0,\infty), && (2.22)\\ u(x,0) &= 0, \quad u_t(x,0) = \psi(x) \quad x \in (0,L). && (2.23) \end{aligned}$$

Clearly, we can repeat steps (2.11)-(2.14) from above. Different situation arises, when the boundary conditions (2.22) are applied to solve (2.14). The solutions of (2.14) satisfying conditions (2.22) are obtained in the form

$$X_n(x) = \cos\left(\frac{n\pi x}{L}\right), \quad \lambda_n = \left(\frac{n\pi x}{L}\right)^2, \quad n = 1, 2, 3, \dots \ . \tag{2.24}$$

Solution of equation (2.13) satisfying conditions (2.23) are written in the full form

$$T_n(t) = a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right), \quad n = 1, 2, 3, \dots \ . \tag{2.25}$$

Solution $u(x,t)$ of our equation is then set up in the form

$$u(x,t) = \sum_{n=1}^{\infty} \cos\left(\frac{n\pi x}{L}\right) \left[a_n \cos\left(\frac{n\pi t}{L}\right) + b_n \sin\left(\frac{n\pi t}{L}\right)\right], \quad n = 1, 2, 3, \dots \ . \tag{2.26}$$

By application of the same ideas as in the previous case, we finally get solution in the form

$$u(x,t) = \sum_{n=1}^{\infty} b_n \cos\left(\frac{n\pi x}{L}\right) \sin\left(\frac{n\pi t}{L}\right), \quad n = 1, 2, 3, \dots \ , \tag{2.27}$$

with coefficients b_n computed by (2.18). Again, final solution $u(x,t)$ is depending on the selection of initial velocity $\psi(x)$.

For the fixed choice of $n = k$ together with proper choice of $\psi(x)$ we obtain the k-th component of the sum (with b_k computed by (2.18)):

$$u_k(x,t) = b_k \cos\left(\frac{k\pi x}{L}\right) \sin\left(\frac{k\pi t}{L}\right). \tag{2.28}$$

The detailed solution of this case and also previous case together with different choice of boundary and initial conditions can be found, for example, in [22].

Vibrating drum (membrane)

The previous considerations have us led to one-dimension solution. Solution of the wave equation in more dimensions is not so easy to find.

Let us present the idea of the solution from [65] (Wave equation: Several space dimensions): *"The one-dimensional initial-boundary value theory may be extended to an arbitrary number of space dimensions. Consider a domain D in m-dimensional x space, with boundary B. Then the wave equation is to be satisfied if x is in D and $t > 0$. On the boundary of D, the solution u shall satisfy*

$$\frac{\partial u}{\partial n} + au = 0,$$

where n is the unit outward normal to B, and a is a non-negative function defined on B. The case where u vanishes on B is a limiting case for variable a approaching infinity. The initial conditions are

$$u(0, x) = f(x), \quad u_t(0, x) = g(x),$$

where f and g are defined in D. This problem may be solved by expanding f and g in the eigenfunctions of the Laplacian in D, which satisfy the boundary conditions. Thus the eigenfunction v satisfies

$$\triangle v + \lambda v = 0,$$

in D, and

$$\frac{\partial v}{\partial n} + av = 0,$$

on B.

*In the case of two space dimensions, the eigenfunctions may be interpreted as the **modes of vibration of a drumhead** stretched over the boundary B. If B is a circle, then these eigenfunctions have an angular component that is a trigonometric function of the polar angle θ, multiplied by a Bessel function (of integer order) of the radial component. If the boundary is a sphere in three space dimensions, the angular components of the eigenfunctions are spherical harmonics, and the radial components are Bessel functions of half-integer order.*

Previous considerations have us led to following remarks.

Remark 2.1.3 (Laplace operator on a disk)
Solution of the vibrating drum problem is, at any point in time, an eigenfunction of the Laplace operator on a disk.

Remark 2.1.4 (Vibration Problem)
Later, as we will refer to the vibration problem (in infinite dimensional space), we will always suppose the solution of wave equation (2.6) in corresponding space dimension, together with some boundary condition (Dirichlet or Neumann).

2.1.2 Unification of the Theory

At the end of this section, let me bring some recapitulation of used notations and ideas. So far, we have used the notation eigenvalue problem, eigenvectors, eigenfunctions, wave equation and Laplace operator (often called *Laplacian*).

Next Section 2.3 is based on results of [6], where the notation Laplace-Beltrami operator is used. In differential geometry, the **Laplace-Beltrami operator** is generalization of the standard **Laplace operator (Laplacian)**; it operates on functions defined on surfaces in Euclidean space and, more generally, on Riemannian and pseudo-Riemannian manifolds. Like the Laplace operator, the Laplace-Beltrami operator is defined as the divergence ($\nabla\cdot$) of the gradient (∇f).

Biyikoglu, Leydold and Stadler [6] explain relationship between eigenvectors of Laplacian and eigenfunctions of Laplace-Beltrami operator by this way: *"From a more formal point of view, **Laplacian eigenvectors** are the natural discretization of **eigenfunctions of Laplace-Beltrami operator** on manifolds. Surprisingly, some of their properties in the discrete case are reminiscent of corresponding results in the continuous setting, but often there are subtle differences which we found interesting enough to explore in some detail."*

The essential thing of following sections is to identify the Laplace-Beltrami operator (in continuous, infinite dimension space) with the standard **Laplacian matrix of graph** (discrete, finite dimension space). The Laplacian matrix of graph is often called as *graph Laplacian*.

Espetially considering the eigenvalue problem in both formulation, we would like to identify, in certain case, the eigenfunctions of the Laplace-Beltrami operator with eigenvectors of Laplacian matrix of graph.

Considering the Laplacian matrix of graph as a matrix representation of a particular case of the **discrete Laplace operator** (see Subsection 2.3.1) or as an approximation to the continuous Laplacian is well known. Viewing of the Laplacian matrix of graph as an approximation to the continuous Laplacian can be interpreted as a matrix form of an approximation to the negative Laplace operator obtained by the *finite difference method*. In this interpretation, every graph vertex is treated as a grid point, the local connectivity of the vertex determines the finite difference approximation stencil at this grid

point, the grid size is always one for every edge, and there are no constraints in any grid points, which corresponds the case of the homogeneous Neumann boundary condition, i.e., "free" boundary. For better understanding, see Subsection 2.3.1 (Finite differences).

2.2 Basic Notations and Definitions

Let us consider a simple, undirected graph $G = (V, E)$ as a set V of vertices together with a set E of edges, which are two–element subsets of V. An edge is related with two vertices, and the relation is represented as unordered pair of the vertices with respect to the particular edge. Further, we consider a simple graph G without loops and multiple edges.

Let us denote P_n as the path on n vertices with $n-1$ edges, C_n as the cycle on n vertices with n edges, K_n as the complete graph on n vertices, and $K_{m,n}$ ass the complete bipartite graph with m vertices in the one part and with n vertices in the second part. The definitions of these types of graphs can be found for example in [66]. Below, we give only these definitions that are essential further in this text.

Definition 2.2.1 (Path, Shortest path)
The path P_n *in a graph* G *is a sequence of vertices* $(v_1, v_2, \ldots, v_n)$, $v_1, v_2, \ldots, v_n \in V(G)$, *such that all edges* $v_i v_{i+1}$ *are in the graph* G *for all* $i = 1, 2, \ldots, n-1$.

The shortest path *between vertices* v_1, v_k *is such a path, for which the number of edges between the vertices* v_1, v_k *is minimum. This path is usually denoted as* (v_1, v_k)*-path.*

Definition 2.2.2 (Walk)
A walk of length k *is a sequence of nodes* $(v_1, v_2, \ldots, v_k)$ *of the given graph, such that the edges* $v_i v_{i+1}$ *are in the mesh for all* $i = 1, 2, \ldots, k-1$.

The walk starts at v_1 *and ends at* v_k *if the first node in the sequence is* v_1 *and the last node is* v_k. *A walk between nodes* v_1 *and* v_k *is called* (v_1, v_k)-walk.

Definition 2.2.3 (Distance)
By $\mathrm{dist}(u, v)$ *we denote the* distance *of the vertices* u *and* v *in a given graph* G, *i.e. the length of the shortest* (u, v)*-path.*

Definition 2.2.4 (Eccentricity)
The eccentricity *of a vertex* u *in graph* G *is the maximum distance to some vertex* v *in the graph:* $\mathrm{ecc}(u) = \max_{v \in V(G)} \mathrm{dist}(u, v)$.

For purposes of this section, we will use the term I for the matrix with ones on the main diagonal and zeros other-where and the term J for the matrix of all ones. Definition of Principal matrix follows.

Definition 2.2.5 (Principal submatrix)
Principal submatrix *of an arbitrary (graph) matrix* A *arises from the matrix* A *by deleting some chosen rows and corresponding columns (corresponding to chosen vertex/vertices).*

2.3 Eigenvalues and Eigenvectors of Graphs

In Subsection 2.3.1, several known types of matrices corresponding to a given graph are presented with special emphasis to the Laplacian matrix of graph. The proper definition of Laplacian matrix is anticipated with less obvious insight to Laplace operator, rather discrete Laplace operator.

Some known results about graph-linear approach to eigenvalues of graphs are presented in Subsection 2.3.2.Special attention is devoted to Fiedler's based contribution to spectral graph theory (Subsection 2.3.3) and to corresponding partitioning of graphs (Subsection 2.3.4).

Some interlacing properties are mentioned in Subsection 2.3.5, to use these results in Section 4.1 and also introduction about Cartesian products of graphs is presented (Subsection 2.3.6).

2.3.1 Matrix Representations of a Graph

In following definitions, some basic types of matrices are defined, based on a given graph $G(V, E)$ with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$ and edge set $E(G) = \{e_1, e_2, \ldots, e_m\}$. The (two) following definitions are well known.

Definition 2.3.1 (Adjacency matrix)
Let G be a loop-less graph with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$ and edge set $E(G) = \{e_1, e_2, \ldots, e_m\}$. The adjacency matrix of G, denoted by $A(G)$, is the n-by-n matrix in which entry a_{ij} is the number of edges in G with end-vertices v_i, v_j.

For a simple graph with no loops, the adjacency matrix has zeros on the diagonal. For an undirected graph, the adjacency matrix is symmetric.

In order to specify incidence matrix M we need to set an arbitrary but fixed orientation (direction) for each edge $e = v_i v_j$.

Definition 2.3.2 (Incidence matrix)
M is an $(|E| \times |V|)$ matrix and has entries $m_{ev} = -1$ if v is the starting vertex of the edge e, $m_{ev} = 1$ if v is the ending vertex of e, and $m_{ev} = 0$ otherwise, i.e., if v is not in e.

The ideas of Biyikoglu, Leydold, and Stadler ([6]) follow. Let us now consider a *real-valued function* f over $V(G)$, $f : V(G) \to \mathbb{R}$ (this is simply a vector indexed by the vertices of G), called *labelling*. Moreover, let us consider the incidence matrix M of G as a matrix operator.

Map $f \mapsto Mf$ is known as the *co-boundary mapping* of the graph G. Its value $(Mf)(e)$ at a given edge e is the difference of the values of f at the

two end-points of edge e (considering the same orientation as setted up for construction of M). Therefore the incidence matrix M is a kind of difference or "discrete differential" operator[4] on G.

Let us now consider an oriented path P_n with vertices $x_0, x_1, \ldots, x_{n-1}$ and edges $e_1, e_2 \ldots, e_{n-1}$; edge e_i starts at vertex x_{i-1} and ends at vertex x_i. We can now define the "2nd derivative" of f in the vertex x_i:

$$\begin{aligned}(\partial f)(x_i) &= (Mf)(e_{i+1}) - (Mf)(e_i) \\ &= (f(x_{i+1}) - f(x_i)) - (f(x_i - f(x_{i-1})) \\ &= f(x_{i+1}) - 2f(x_i) + f(x_{i-1}).\end{aligned} \tag{2.29}$$

Note that $(\partial f)(x)$ in Equation (2.29) holds only for paths and it is independent on the orientation on G. It seems natural to consider the sum over these "2nd derivatives" as a (discrete) *Laplace-Beltrami operator*[5] and generalize this definition for an arbitrary graph:

$$\begin{aligned}(\triangle f)(x_i) &= \sum_{\text{passes through } x_i} (\partial f)(x_i) \\ &= \sum_{\text{passes through } x_i} f(x_{i+1}) - 2f(x_i) + f(x_{i-1}) \\ &= \sum_{x_k x_i \in E(G)} (f(x_k) - f(x_i)).\end{aligned} \tag{2.30}$$

The $x_k x_i \in E(G)$ means that vertex x_k is adjacent to the vertex x_i. The sum in above Equation (2.30) goes over all adjacent vertices x_k to the vertex x_i.

The above definition of $\triangle f$ from [6] is unclear, it allows several explanations. For example if $\deg(x_i)$ is odd, should be there any edge counted twice or not? I propose another definition of $\triangle f$ (which has the same meaning).

$$(\triangle f)(x_i) = \sum_{\text{output edges from } x_i} (Mf)(e_{out}) - \sum_{\text{input edges to } x_i} (Mf)(e_{in}). \tag{2.31}$$

Here, the orientation of edges is the same as setted up for construction of incidence matrix M.

[4] In [6], the authors denote the symbol for matrix M by symbol ∇. I do not preserve this notation to prevent confusion with the notation for the standard gradient.

[5] In differential geometry, the *Laplace-Beltrami operator* is a generalization of the standard *Laplace operator*; it operates on functions defined on surfaces in Euclidean space and, more generally, on Riemannian and pseudo-Riemannian manifolds. Like the Laplacian, the Laplace–Beltrami operator is defined as the divergence $(\nabla\cdot)$ of the gradient (∇f), $\triangle f$ = div grad f, i.e. $\triangle f = \nabla \cdot \nabla f$, and it is a linear operator taking functions into functions.

In graph theory literature, it is customary to define the *Laplace operator*, Laplace-Beltrami operator respectively, *as a mapping* $\mathcal{L} : \mathbb{R}^{|V|} \to \mathbb{R}^{|V|}$ with the negative sign:

$$(\mathcal{L}f)(x_i) = (-\triangle f)(x_i) = \sum_{x_i x_k \in E(G)} (f(x_i) - f(x_k)). \tag{2.32}$$

For the function $\mathcal{L}f$ it can be also written

$$(\mathcal{L}f)(x_i) = \sum_{x_i x_k \in E(G)} (f(x_i) - f(x_k)) = d(x_i)f(x_i) - \sum_{x_i x_k \in E(G)} f(x_k), \tag{2.33}$$

where $d(x_i)$ denotes the degree of the vertex x_i.

From an algebraic point of view the *Laplacian matrix* of G could be define

$$L = M^T M. \tag{2.34}$$

Another way how to define the Laplacian matrix is the following definition, which is more common.

Definition 2.3.3 (Laplacian matrix)
The Laplacian matrix $L(G)$ *of* G *is an* $n \times n$ *symmetric matrix with one row and one column for each vertex defined by*

$$L_{ij}(G) = \begin{cases} d(i), & \text{if } i = j \\ -1, & \text{if } i \neq j \text{ and } ij \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

for $i, j = 1, ..., n$, *where* $d(i)$ *is the vertex degree of node* i.

As L is a symmetric matrix, L_{ij} is independent of the orientation of the edges.

Remark 2.3.4 (see [6])
The definition of Laplacian matrix is equivalent to equation (2.32), (2.33) respectively. The identity $(\mathcal{L}f)(x) = (Lf)(x)$ *yields the Laplacian matrix* L *can be viewed as a proper discretization of the usual Laplace-Beltrami differential operator.*

This remark is adopted from [6]. To avoid misunderstanding, I would rather use different notation. At the beginning of this section, the real-valued function f over $V(G)$, $f : V(G) \to \mathbb{R}$ is considered. As matrix L is considered as an $n \times n$ matrix ($n = |V|$), we obtain f as an element of

the vector space $\mathbb{V} \stackrel{\text{def}}{=} \mathbb{R}^{|V|}$ ($f \in \mathbb{V}$). This consideration is natural, we simply identify the function value $f(v_i)$ with the i-th component $f[i]$ of vector f. We should rather write $(\mathcal{L}f)(x) = (L \cdot f)[x]$ where $L \cdot f$ is simply the matrix-vector multiplication.

Matrices $L(G)$ and $A(G)$ are related via $L(G) = D(G) - A(G)$, $D(G)$ is an $n \times n$ diagonal matrix where d_{ii} is the i-th diagonal element of $D(G)$ representing the vertex degree of node i in graph G, and $d_{ij} = 0$ for $i \neq j$.

Based on Equation (2.33), we can define the generalized Laplacian matrix.

Definition 2.3.5 (Generalized Laplacian matrix, [6])
We call a symmetric matrix H a generalized Laplacian matrix *of $G = (V, E)$ if $h_{ij} < 0$ whenever ij is an edge of G and $h_{ij} = 0$ whenever i and j are distinct and not adjacent. There are no constraints on the diagonal entries of H[6]. In algebraic form, H can be consider as a Hamiltonian operator $\mathcal{H}$ of the form*

$$(\mathcal{H}f)(x_i) = \sum_{x_i x_k \in E(G)} w_{x_i x_k}(f(x_i) - f(x_k)) + p(x_i)f(x_i), \qquad (2.35)$$

where the term $w_{x_i x_k}$ denotes the 'weight' of an edge $x_i x_k$, $p(x_i)$ represents certain potential.

Ordinary Laplacian matrix L as well as the negative adjacency matrix $-A$ are of course special cases of the generalized Laplacian.

There are some other definitions of the Laplacian matrix, for example a *normalized Laplacian matrix*, see [13]. In the rest of the text the standard Laplacian matrix will be used, according to Definition 2.3.3, if not said otherwise.

Finite Differences

Partial elliptic differential equations arise in many applications of mathematical physics. One of such examples is the Poisson's equation:

$$-\Delta u = f \text{ in } \Omega, \qquad (2.36)$$

where $u, f \in \mathbb{R}^s$, $\Omega \subseteq \mathbb{R}^s$. Here Δ denotes the classical Laplace operator given as $\Delta u = \sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2}$. Solving these types of equations is a challenging task in numerical mathematics. One approach is to use *finite difference* discretization scheme. At the nodes of the discretization grid the Laplace

[6] Fiedler [28] calls such matrices "essentially non-positive".

operator Δ is approximated by a difference operator Δ_h. For $\Omega \subseteq \mathbb{R}^2$ and a square mesh of width h we get the so-called Five-Point Formula:

$$\Delta_h u(x,y) = \frac{u(x+h,y) + u(x,y+h) + u(x-h,y) + u(x,y-h) - 4u(x,y)}{h^2}. \tag{2.37}$$

Δ_h is therefore the same as the standard Laplacian matrix (defined on the discretization grid) times the constant $-\frac{1}{h^2}$.

The Discrete Dirichlet Problem

As the standard Laplacian matrix corresponds naturally to the Poisson's equation with the Neumann condition, it does not require any other consideration. The Poisson's equation with the Neumann condition is described further in this text (see Section 3.1.1).

Let us now consider the previous Poisson's equation (2.36) with the Dirichlet condition.

$$-\Delta u = f \text{ in } \Omega, \qquad u(x) = 0 \text{ on } \Gamma, \tag{2.38}$$

where Γ is the boundary of the domain $\Omega \subseteq \mathbb{R}^2$. We can imagine this problem as a two-dimensional membrane. The condition on the boundary simply means, that the membrane is fixed on the boundary against motions in direction x. Again, Δ denotes the Laplace-Beltrami operator.

As the discretization of the membrane creates a mesh on Ω, we can consider this mesh as a graph G. The discretization yields a Laplacian matrix L of graph G. The eigenvalues of L correspond to frequencies of the membrane (see Section 2.1, especially Remark 2.1.2 and Remark 2.1.3).

Let us cite the ideas in [6]: *"We need a notion of a graph with boundary for defining discrete analogues of Dirichlet boundary conditions. Of course, graphs do not have boundaries by themselves. Starting from a graph $G(V,E)$ we may, however, consider the induced sub-graph $G(\overline{V})$ on a subset $\overline{V} \subseteq V$, considering $V \setminus \overline{V}$ as the boundary of $G(\overline{V})$ on which the constraint $u(x) = 0$ is enforced. We denote this boundary by ∂V. Formally we can define a graph with boundary as a graph $G(\overline{V} \cup \partial V, \overline{E} \cup \partial E)$ where $\overline{V}$ denotes the set of interior vertices and ∂V the set of boundary vertices. The set of edges between interior vertices are called interior edges and denoted by $\overline{E}$; edges between $\overline{V}$ and ∂V are called boundary edges and denoted by ∂E. Edges between boundary vertices do not make sense in our setting and are thus deleted. It must be noted here that a graph with boundary is called connected if the graph induced by its interior vertices, $G(\overline{V})$, is connected.*

If we restrict ourselves to solutions u of the Dirichlet problem on a graph $G(\overline{V} \cup \partial V, \overline{E} \cup \partial E)$ with boundary ∂V we have to look for a function u which

vanishes on all boundary vertices, i.e. $u(x) = 0$ *for* $x \in \partial V$*, and which satisfies for all interior vertices* $x \in \overline{V}$

$$\begin{aligned}(\mathcal{L}u)(x) &= (-\triangle u)(x) = \sum_{xy \in E(G)} (u(x) - u(y)) \\ &= \sum_{y \in V} L_{xy} u(y) = \sum_{y \in \overline{V}} L_{xy} u(y) \\ &= f(x).\end{aligned} \tag{2.39}$$

for some eigenvalue λ*. Thus the Dirichlet problem can be reduced to a matrix eigenspace problem for* $G(\overline{V})$*. The corresponding Dirichlet (Laplacian) matrix* $\overline{L}(G)$ *can be derived from the graph Laplacian* $L(G)$ *simply by deleting all rows and columns that correspond to boundary vertices, i.e., by using the principal submatrix corresponding to interior vertices. Compared to the 'free' graph Laplacian* $L(G(\overline{V}))$ *on the graph induced by its interior vertices,* $G(\overline{V})$*, the Dirichlet matrix differs just by an additional contribution* $p(x)$ *in the diagonal elements:*

$$\overline{L}(G) = L(G(\overline{V})) + P, \tag{2.40}$$

where P *is a diagonal matrix whose entries are* $\{p_{ii}\} = p(x_i) = |\{x_k : x_k x_i \in \partial E\}|$*."*

2.3.2 Linear Algebra of Real Symmetric Matrices

There is a lot of claims about eigenvalues of graphs, see for example [13]. Here, we will show only some of them. It is essential to be careful with which matrix type the particular proof deals.

In the following, we present several theorems from linear algebra. Proof of them can be found for example in [66].

Definition 2.3.6 (Characteristic polynomial)
The eigenvalues of a matrix A *are the numbers* λ *such that* $Ax = \lambda x$ *has a non-zero solution vector* x*. Each solution is an eigenvector associated with* λ*. These* λ *are roots* $\lambda_1, \lambda_2, \ldots, \lambda_n$ *of the characteristic polynomial* $\phi(G, \lambda) = det(\lambda I - A)$*.*

The spectrum is a list of distinct eigenvalues with their multiplicities $m_1, \ldots, m_l$*; we write* $\mathrm{Spec}(G) = \{\lambda_1^{[m_1]}, \lambda_2^{[m_2]}, \ldots, \lambda_l^{[m_l]}\}$*.*

We will consider, that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{k-1} < \lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+m_k-1} < \lambda_{k+m_k} \leq \cdots \leq \lambda_n$. Here, the case when λ_k has the multiplicity m_k is considered.

Theorem 2.3.7 (Spectral Theorem, [66])
A real symmetric $n \times n$ matrix has real eigenvalues and n orthonormal eigenvectors.

Both adjacency and Laplacian matrices are real and symmetric, so their eigenvalues are real values. Moreover, eigenvalues of the Laplacian matrix are non-negative (by the Gershgorin circle theorem, see for example [29]).

The quadratic form $\langle f, Lf \rangle$ of the graph Laplacian (see equation (2.32)) can be computed via Greens formula as

$$\langle f, Lf \rangle = \sum_{x_i, x_k \in V} L_{x_i x_k} f(x_i) f(x_k) = \sum_{x_i x_k \in E} (f(x_i) - f(x_k))^2. \quad (2.41)$$

This equality immediately shows that the quadratic form of the graph Laplacian is a non-negative operator, i.e., all eigenvalues are greater than or equal to zero. The Laplacian matrix is positive semi-definite.

As we know, that the adjacency and the Laplacian matrix are related through degrees of vertices, we can set up following lemma.

Lemma 2.3.8 ([6])
Let G be a k-regular graph. If the adjacency matrix $A(G)$ has eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, then the Laplacian $L(G)$ has eigenvalues $k-\lambda_1, k-\lambda_2, \cdots, k-\lambda_n$.

Proof. If G is k-regular, then $L(G) = D(G) - A(G) = kI - A(G)$. From eigenvalue equation (2.1) $A(G) \cdot v = \lambda \cdot v$. Substituting $A(G) = kI - L(G)$, we can directly write

$$\begin{aligned} (kI - L(G)) \cdot v &= \lambda \cdot v \\ L(G) \cdot v &= -\lambda \cdot v + k \cdot v \\ L(G) \cdot v &= (k - \lambda) \cdot v. \end{aligned} \quad (2.42)$$

Therefore, $(k - \lambda)$ is an eigenvalue of $L(G)$ and every eigenvector v of $A(G)$ with eigenvalue λ is an eigenvector of $L(G)$ with eigenvalue $k - \lambda$. □

Fan R. K. Chung [13] and B. Mohar [52] list basic properties of the Laplacian matrix and present a survey of know results about the spectrum of $L(G)$ with special emphasis on λ_2 and its relation to numerous graph invariants. Their work was based on ideas of M. Fiedler, which is introduced in Subsection 2.3.3.

Let us have a look on another important property of $L(G)$. This is standard result, presented e.g. in [1] and it is often used for various proofs of claims about graph matrices.

Lemma 2.3.9 ([66])
If $f(x) = x^T Ax$, where A is a real symmetric matrix, then f attains its maximum and minimum over unit vertors x at eigenvectors of A, where $f(x)$ equals to the corresponding eigenvalues.

The Rayleigh Quotient

Definition 2.3.10 (Rayleigh Quotient for (graph) Laplacian matrix)
Rayleigh quotient $R(f)$ of a function $f : V \to \mathbb{R}$ with respect to a Laplacian matrix L is a number defined by the fraction

$$R(f) = \frac{\langle f, Lf \rangle}{\langle f, f \rangle}. \tag{2.43}$$

For graph Laplacian L this can equivalently be written as

$$R(f) = \frac{\sum_{x_i x_k \in E} (f(x_i) - f(x_k))^2}{\sum_{x_i \in V} f(x_i)^2}. \tag{2.44}$$

Here again, f is a function from the vector set V to $\mathbb{R}$ ($f : V \to \mathbb{R}$). For further formulations f is considered to be an element of the vector space $\mathbb{V} \stackrel{\text{def}}{=} \mathbb{R}^{|V|}$ ($f \in \mathbb{V}$). Simply, we identify the function value $f(v_i)$ with the i-th component $f[i]$ of an vector f.

If f happens to be k-th eigenvector $\vec{v}_k$ of L, then the Rayleigh quotient for f is equal to λ_k. In fact, each eigenvalue of L is an extreme value of the Rayleigh quotient over an appropriate subspace of $\mathbb{V}$.

Theorem 2.3.11 (Rayleigh's Principle, [56])
Let $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathbb{V}$ ($\mathbb{V}$ is spanned by the eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$) denote orthogonal eigenvectors corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of a Laplacian L. Let $\{\vec{v}_1.\vec{v}_2, \ldots, \vec{v}_{k-1}\}$ be the set of the first $k-1$ eigenvectors, $\mathbb{V} \supseteq V_{k-1} = \langle \vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{k-1} \rangle$ (V_{k-1} is spanned by the eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{k-1}$).

For each $1 \leq k \leq n$,

$$\lambda_k = \min_{\vec{v} \perp V_{k-1}} \frac{\vec{v}^T A \vec{v}}{\vec{v}^T \vec{v}}.$$

Let us remark, that the notation $\vec{v} \perp V_{k-1}$ denotes the same as $\vec{v} \in V_{k-1}^{\perp}$, where $\mathbb{V} \supseteq V_k^{\perp} = \langle v_k, \ldots, v_n \rangle$ ($n = |V|$) is the orthogonal complement of V_{k-1}.

In general, the idea is written as a well known *Min-max theorem* (Courant–Fischer theorem).

Theorem 2.3.12 (Min-max theorem, [6])
Let W_k and $W_{k-1}^{\perp}$ denote the sets of subspaces of $\mathbb{R}^n$ of dimension at least k and of co-dimension at most $k-1$, respectively. Then

$$\lambda_k = \min_{W \in W_k} \max_{o \neq \vec{v} \in W} \frac{\vec{v}^T A \vec{v}}{\vec{v}^T \vec{v}} = \max_{W \in W_{k-1}^{\perp}} \min_{o \neq \vec{v} \in W} \frac{\vec{v}^T A \vec{v}}{\vec{v}^T \vec{v}}. \tag{2.45}$$

If matrix A is the graph Laplacian matrix, we can also use the corresponding notion as used before

$$\lambda_k = \min_{W \in W_k} \max_{o \neq f \in W} \frac{\langle f, Lf \rangle}{\langle f, f \rangle} = \max_{W \in W_{k-1}^{\perp}} \min_{o \neq f \in W} \frac{\langle f, Lf \rangle}{\langle f, f \rangle}. \tag{2.46}$$

Take the standard Laplacian matrix L. Considering subspaces $V_k, V_{k-1}^{\perp} \subseteq \mathbb{V}$ spanned by eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$, or $\vec{v}_k, \vec{v}_{k+1}, \ldots, \vec{v}_n$ respectively, as in Theorem 2.3.11, we can write the second part of Equation 2.45 for several important eigenvalues.

Suppose $k = 1$, i.e. we are interesting in the smallest eigenvalue

$$\lambda_1 = \max_{\vec{v} \in \mathbb{V}} \min_{o \neq \vec{v} \in \mathbb{V}} \frac{\vec{v}^T L(G) \vec{v}}{\vec{v}^T \vec{v}} = \min_{\vec{v} \in \mathbb{V}} \frac{\vec{v}^T L(G) \vec{v}}{\vec{v}^T \vec{v}}. \tag{2.47}$$

The equality arises when $\vec{v}$ is the eigenvector corresponding to the smallest eigenvalue λ_1. Considering the Laplacian matrix, it has the first eigenvalue $\lambda_1 = 0$ (the Laplacian matrix is the positive semi-definitive matrix with a one-dimensional kernel – see Section 3.1). The corresponding eigenvector is a constant eigenvector, for example the vector of all ones $(1, 1, ..., 1)$.

If $L(G)$ is the Laplacian matrix, the second smallest eigenvalue λ_2 is given by

$$\lambda_2 = \max_{V \in V_1^{\perp}} \min_{o \neq v \in V} \frac{\vec{v}^T L(G) \vec{v}}{\vec{v}^T \vec{v}} = \min_{\vec{v} \perp (1,1,\ldots,1)} \frac{\vec{v}^T L(G) \vec{v}}{\vec{v}^T \vec{v}}. \tag{2.48}$$

Here, V_1 is the subspace of $\mathbb{V}$ spanned by the first eigenvector $v_1 = (1, 1, ..., 1)$. Moreover, V is given as the whole $V_1^{\perp}$ (i.e. maximal). The symbol $\vec{v} \perp (1, 1, ..., 1)$ denotes the same as $\vec{v} \in V_1^{\perp}$ and it means that the formula operate on the vector space orthogonal to the first eigenvector.

Similary, using the first part of Equation 2.45, we can compute the highest eigenvalues. Suppose $k = n$, i.e. we are interesting in the highest eigenvalue.

$$\lambda_n = \min_{\mathbb{V}} \max_{o \neq \vec{v} \in \mathbb{V}} \frac{\vec{v}^T L(G) \vec{v}}{\vec{v}^T \vec{v}} = \max_{\vec{v} \in \mathbb{V}} \frac{\vec{v}^T L(G) \vec{v}}{\vec{v}^T \vec{v}}. \tag{2.49}$$

The equality arises when $\vec{v}$ is the eigenvector corresponding to the highest eigenvalue λ_n. Again, using the orthogonalization to this eigenvector, $\lambda_{n-1}, \lambda_{n-2}, \ldots$, etc. can be computed.

The Perron-Frobenius Theorem

The next well known and useful theorem is the *Perron-Frobenius theorem*. There are many forms and applications of this theorem. After some necessary preliminaries, one of its basic for will be introduced.

A real matrix $M = (m_{ij})$ is called *non-negative* if each of its entries is non-negative. A non-negative, square matrix M is called *irreducible* if for each pair (i, j), there exists a non-negative integer p such that the $(i, j)^{th}$ entry of M^p is strictly positive. Since the $(i, j)^{th}$ entry of the p^{th} power of the adjacency matrix of a graph G is equal to the number of edge sequences of length p connecting vertex i to vertex j, it is clear that the adjacency matrix of a connected graph is irreducible.

The spectral radius $\rho(M)$ of a matrix or a bounded linear operator is the supremum among the absolute values of the elements in its spectrum. Considering $\lambda_1, \lambda_2, \ldots, \lambda_n$ as (real or complex) eigenvalues of a matrix M, the spectral radius is computed as

$$\rho(M) = \max_{i=1,2,\ldots,n} (|\lambda_i|). \tag{2.50}$$

Theorem 2.3.13 *(Perron-Frobenius)*
Let A and B be real symmetric irreducible non-negative $n \times n$ matrices. Then

(i) the spectral radius $\rho(A)$ is a simple eigenvalue of A. If v is an eigenvector corresponding to $\rho(A)$, then no entries of v are zero, and all have the same sign.

(ii) If moreover $A - B$ is non-negative, then $\rho(B) \leq \rho(A)$, with equality if and only if $B = A$.

Let us apply this theorem to get a statement about the eigenvalue counted in $\rho(M)$ and its eigenvector of matrix M. For each i, the row sum r_i of a matrix $M = (m_{ij})$ is given by

$$r_i = \sum_{j=1,2,\ldots,n} m_{ij}.$$

Theorem 2.3.14 *Let M be a non-negative, square matrix, and suppose M is irreducible. Let r_{min} and r_{max} be the minimum and maximum row sums of M, respectively. There is a unique eigenvector v of M all of whose entries are positive. The eigenvalue λ corresponding to v is the largest eigenvalue of M and satisfies $r_{min} \leq \lambda \leq r_{max}$.*

The proof of this theorem can be found, e.g. in [1].

Adjacency matrix $A(G)$ defined in Definition 2.3.1 of connected graph G is a special case of an irreducible non-negative matrix M. We refer to the positive eigenvector corresponding to the largest eigenvalue of $A(G)$ as to the *Perron vector*.

2.3.3 Fiedler Based Contributions to Graph Theory

Let me introduce several amazing theorems by Miroslav Fiedler [27], [28]. These theorems were established in 70's, and they are still studied and useful for a lot of applications, for example in spectral graph partitioning.

For purposes of this subsection, $L(G)$ is referred to as the Laplacian matrix of a graph G according to Definition 2.3.3.

Lemma 2.3.15 (Algebraic connectivity)
Let $n \geq 2$ and let the eigenvalues of $L(G)$ be ordered $\lambda_1 = 0 \leq \lambda_2 \leq ... \leq \lambda_n$. An eigenvector corresponding to λ_1 is a vector of all ones. The multiplicity of λ_1 is equal to the number of connected components of the graph G.

The second smallest eigenvalue λ_2 is greater then zero iff G is connected. Fiedler [27] called this number *algebraic connectivity* of graph and he marked it as $a(G)$. It follows, that the rank of $L(G)$ is $n - k(G)$, where $k(G)$ is the number of connected components of G.

Fiedler showed that $a(G)$ is closely related to $v(G)$ and e(G), the vertex and edge connectivities of G, respectively.

Lemma 2.3.16 (Fiedler, 1973 [27])
Let G be a graph, let G_k arise from G by removing k vertices from G and all incident edges, then

$$a(G_k) \geq a(G) - k. \tag{2.51}$$

Theorem 2.3.17 (Fiedler, 1973 [27])
Let G be a graph that is not complete. Then

$$a(G) \leq v(G) \leq e(G). \tag{2.52}$$

Fiedler has also investigated graph-theoretical properties of the eigenvector corresponding to λ_2, called second eigenvector. The coordinates of this eigenvector are assigned to the vertices of G in a natural way and can be considered as valuation of the vertices of G. Fiedler called this valuation *characteristic valuation of G*. In honour of M. Fiedler, this eigenvector is often called the *Fiedler eigenvector*. Some of his results follow.

Theorem 2.3.18 ("Fiedler's tree theorem")
Let T be a (weighted) tree on n vertices, labelled $1, 2, \ldots, n$, with Laplacian matrix L and algebraic connectivity a. Let v be an eigenvector of L associated with a. Then exactly one of the following cases occurs:

*(i) Some entry of v is 0. In this case the sub-graph of T induced by the set of vertices corresponding to the zero entries in v is connected. Moreover, there is a unique vertex (*the characteristic vertex*) k such that $v[k] = 0$ and k is adjacent to a vertex m with $v[m] \neq 0$. The entries of v are either increasing and concave, decreasing and convex, or identically 0 along any path in T which starts at k.*

*(ii) No entry of v is 0. In this case there is a unique pair of vertices i and j (*the characteristic edge*) such that i and j are adjacent in T with $v[i] > 0$ and $v[j] < 0$. Furthermore, the entries of v are increasing and concave along any path in T which starts at i and does not contain j, while the entries of v are decreasing and convex along any path in T which starts at j and does not contain i.*

The results of this theorem are used in next Subsection 2.3.4. Following theorem is essential for the spectral graph partitioning. It is used to the spectral bisection of graphs.

Theorem 2.3.19 ([28])
Let G be a finite connected graph with n vertices $1, 2, ..., n$. Let $y = (y_i)$ be a characteristic valuation of G. For any $r \geq 0$, let

$$M(r) = \{i \in N | y_i + r \geq 0\}.$$

Then the subgraph $G(r)$ induced by G on $M(r)$ is connected.

Remark 2.3.20 *A similar statement holds for $r \leq 0$ and the set $M'(r)$ of all those indices i for which $y_i + r \leq 0$.*

From these results it simply follows that by dividing a finite connected graph into two parts at least one of the parts is connected (the connectivity might be broken in those vertices for which $y_i + r = 0$). According to Theorem 2.3.19, the first part consists of sub-graph $G(r)$ induced by G on $M(r)$ and Fiedler has proved, that this part is always connected. The second part consists from the complement of $G(r)$ to G (exceptthe edges on the cut) and it has not to be connected.

The proofs of these results and also of the following corollary can be found in [28].

Corollary 2.3.21 ([28])
Let G be a valuated connected graph with vertices 1,2,...n, let $y = (y_i)$ be a characteristic valuation of G.

(i) If c is a number such that $0 \leq c < max(y_i)$ and $c \neq y_i$ for all i then the set of all those edges (i, k) of G for which $y_i < c < y_k$ forms a cut C of G. If $N_1 = \{k \in N | y_k > c\}$ and $N_2 = \{k \in N | y_k < c\}$ then $N = (N_1, N_2)$ is a decomposition of N corresponding to C and the subgraph $G(N_2)$ is connected.

(ii) If $y_i \neq 0$ for all $i \in N$ the set of all alternating edges, i.e. edges (i, k) for which $y_i y_k < 0$, forms a cut C of G such that both subgraphs of G are connected.

2.3.4 Nodal Domains

The idea of (graph) nodal domains is adopted from Courant's theorem for elliptic operators on manifolds, later called as the Courant's nodal domain theorem [11]. *"Given the self-adjoint second order differential equation $\triangle u + \lambda \rho u = 0$ $(\rho > 0)$ for a domain G with arbitrary homogeneous boundary conditions; if its eigenfunctions are ordered according to increasing eigenvalues, then the nodes of the n-th eigenfunction u_n divide the domain into no more than n subdomains. No assumptions are made about the number of independent variables."* These *subdomains* are then referred to as *nodal domains*.

Following the ideas of Theorem 2.3.19, of Remark 2.3.20 and choosing $r = 0$, the subgraph induced by non-positive vertices (i.e., vertices with non-positive function values) and the subgraph induced by non-negative vertices are obtained. According to the theorem, both such subgraphs are connected. In terminology of nodal domains, an eigenfunction corresponding to the second eigenvalue has exactly two *weak nodal domains* (see Definition 2.3.22 below). Fiedler's ideas were extended by a lot of followers, see for example Powers [55] or Davies, Gladwell, Leydold, Stadler [14].

The discrete analog of a "nodal domains" is a maximal connected induced subgraph consisting entirely of positive or negative vertices (i.e., vertices with positive or negative function values of some valuation), which would more appropriately be called a sign graph. The Courant's Nodal Domain Theorem was rewritten into discrete form by Davies, Gladwell, Leydold, Stadler in [14]. At first, let us define several therms.

Definition 2.3.22 (Strong and weak nodal domains, [6])
A positive (negative) strong nodal domain *of a function f on $V(G)$ is a*

maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) > 0$ ($f(v) < 0$).

On the other hand, a positive (negative) weak nodal domain *of a function f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ ($f(v) \leq 0$) that contains at least one nonzero vertex.*

The following theorem by Davies, Gladwell, Leydold, Stadler ([14]) is the discrete analog to the Courant's nodal domain theorem.

Theorem 2.3.23 (Discrete Nodal Domain Theorem, [14])
Let M be a generalized Laplacian of a connected graph with n vertices. Then any eigenfunction f_k corresponding to the k-th eigenvalue λ_k with multiplicity r has at most k weak nodal domains and at most $k + r - 1$ strong nodal domains.

Davies, Gladwell, Leydold, Stadler ([14]) also defined that two different weak (strong) nodal domains D_1 and D_2 of a graph G generated by a function f are *adjacent* if there exist vertices $v_1 \in D_1$ and $v_2 \in D_2$ such that $v_1v_2 \in E(G)$. By the definition, if two different weak (strong) nodal domains are adjacent, then they have opposite signs[7].

Let us return to previous Subsection 2.3.3 based on Fiedler's results. One of the interesting lemmas based on Fiedler's theorems is Lemma 2.3.24 below, first introduced by Kirkland et al. [43] for trees (Perron branches). For purposes of this lemma, several terms have to be established, adopted from [6].

Let W be the characteristic vertex set (i.e. characteristic set that entirely consists of vertices)[8] of some Fiedler vector on a graph G. Then each component D of $G \setminus W$ is a strong sign graph (nodal domain) of f.

The principal submatrix (see Definition 2.2.5) of $L(G)$ corresponding to this component is simply the Dirichlet matrix $\overline{L}(\overline{D})$, see Subsection 2.3.1, where $\overline{D}$ is the graph with boundary that consists of D and adjacent vertices from W as its boundary vertices and the corresponding edges as its boundary edges. If W contains a characteristic edge of a Fiedler vector f we proceed similarly, see [6].

Lemma 2.3.24 *Let f be an eigenfunction corresponding to some eigenvalue λ of $L(G)$, and let D be a geometric nodal domain of f. Then $\overline{\lambda}(D) = \lambda$ with*

[7]The sign of the weak (strong) domain is induced by the signs of the vertices in that domain, i.e. either nonnegative (positive) or nonpositive (negative).

[8]Theorem 2.3.18 has defined the characteristic vertex or the characteristic edge for trees. In general, these can be similarly defined for other types of graphs. In that case, there can arise more vertices, or edges, respectively.

f restricted to the interior vertices of D as its eigenfunction, where $\overline{\lambda}(D)$ *denotes the (Dirichlet) eigenvalue of* $\overline{L}(D)$.

Proof of this lemma can be found, for example, in [6]. Several others theorems following this lemma can be found in [2], [3]. The valuable contributions to the eigenvectors of Laplacian matrix of graph can be found in [49], [50].

2.3.5 Interlacing Properties

In this subsection, let us present some "interlacing" properties of the adjacency and the Laplacian matrix based on the well known Cauchy interlacing theorem. The known results can be found, for example, in [37]. The following theorem is well known and it is often used for various proofs about spectrum of Laplacian matrices.

Theorem 2.3.25 (Cauchy Interlacing Theorem)
Let A be a real $n \times n$ *symmetric matrix and B be an* $(n-1)\times(n-1)$ *principal submatrix obtained by deleting one row and the corresponding column from A. If* $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ *and* $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_{n-1}$ *are the eigenvalues of A and B, respectively, then*

$$\lambda_i \leq \theta_i \leq \lambda_{i+1}, \quad \forall\ i = 1, 2, \ldots, n-1.$$

Cauchy interlacing theorem deals with deleting one row and column. Similar result holds also when r ($r < n$) rows and columns are deleted (proved by a recursive application of Cauchy interlacing theorem). The eigenvalues are then ordered

$$\lambda_i \leq \theta_i \leq \lambda_{i+r}, \quad \forall\ i = 1, 2, \ldots, n-r.$$

Following theorem is the "vertex version" of the Cauchy interlacing theorem (2.3.25). Note, that it holds only for the adjacency matrix of graph, which has zeros on the main diagonal!

Theorem 2.3.26 (Deleting vertex, Adjacency matrix)
Let G be a graph and $G^- = G - v$, *where v is a vertex of G. If* $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ *and* $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_{n-1}$ *are the eigenvalues of* $A(G)$ *and* $A(G^-)$, *respectively, then*

$$\lambda_i \leq \theta_i \leq \lambda_{i+1}, \quad \forall\ i = 1, 2, \ldots, n-1.$$

Cauchy's interlacing does not directly apply to the standard Laplacian matrix since the principal submatrix of a Laplacian matrix may no longer be the Laplacian matrix of a subgraph. However, the following result given by van den Heuvel or by Mohar reflects an edge version of the interlacing property.

Theorem 2.3.27 (Deleting edge, Laplacian matrix)
Let G be a graph and $G^- = G - e$, where e is an edge of G. If $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $0 = \theta_1 \leq \theta_2 \leq \cdots \leq \theta_{n-1}$ are the eigenvalues of $L(G)$ and $L(G^-)$, respectively, then

$$\lambda_i \leq \theta_{i+1} \leq \lambda_{i+1}, \quad \forall\ i = 1, \ldots, n-1.$$

and $\theta_1 = 0$.

Also, there are several new results considering another combinations of deleting vertices/edges of Adjacency/Laplacian matrix. This work has been done by F. Hall, K. Patel, and M. Stewart in 2009-2010 [31] but it has not been published yet.

Theorem 2.3.28 (Deleting vertex, Laplacian matrix)
Let G be a graph of order n and $G^- = G - v$, where v is a vertex of G of degree r. If $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $0 = \theta_1 \leq \theta_2 \leq \cdots \leq \theta_{n-1}$ are the eigenvalues of $L(G)$ and $L(G^-)$, respectively, then

$$\lambda_i \leq \theta_{i+r-1} \leq \lambda_{i+r}, \quad \forall\ i = 1, 2, \ldots, n-r$$

and

$$0 \leq \theta_i \leq \lambda_{i+1}, \quad \forall\ i = 1, 2, \ldots, r-1.$$

The proof of this theorem is based on the knowledge that $L = M^T M$ (Equation (2.34)) and on the application of Cauchy interlacing theorem 2.3.25 on the matrix M when more rows/columns are deleted.

F. Hall, K. Patel, and M. Stewart have proved more types of interlacing theorems, especially on the normalized Laplacian matrix, but for our purposes the presented results will be sufficient. For completeness only, another theorem (by F. Hall, K. Patel, and M. Stewart) is presented.

Theorem 2.3.29 (Deleting vertex, Adjacency matrix)
Let G be a graph and $G^- = G - e$, where e is an edge of G. If $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_n$ are the eigenvalues of $A(G)$ and $A(G^-)$, respectively, then

$$\lambda_{i-1} \leq \theta_i \leq \lambda_{i+1}, \quad \forall\ i = 2, \ldots, n-1,$$

and

$$\theta_1 \leq \lambda_2,\ \lambda_{n-1} \leq \theta_n.$$

2.3.6 Cartesian Product of Graphs

From a graph theoretical point of view the square mesh is a Cartesian product of two paths of given lengths, $P_m \times P_n$. Therefore, it is reasonable to investigate meshes as (generalized) Cartesian products.

The *Cartesian product* $G_1 \times G_2$ of graphs G_1, G_2 is the graph on the vertex set $V(G_1) \times V(G_2)$ (Cartesian product of vertex sets $V(G_1)$, $V(G_2)$). The edge set $E(G_1 \times G_2)$ is constructed as follows. Let $v_1, v_2 \in V(G_1)$ and $u_1, u_2 \in V(G_2)$. Then vertices (v_1, u_1) and (v_2, u_2) are adjacent in $G_1 \times G_2$ if and only if one of the conditions is satisfied: (i) $v_1 = v_2$ and $\{u_1, u_2\} \in E(G_2)$, or (ii) $\{v_1, v_2\} \in E(G_1)$ and $u_1 = u_2$.

From this introduction we can see, that a "grid graph" is a Cartesian product of two paths. This types of graphs play an important role when analyzing two dimensional space meshes. Similarly, also the Cartesian product of three paths $P_k \times P_l \times P_m$ can be defined, which corresponds to a three dimensional space "grid graph".

At this point, let us to recall one of the most interesting theorems about spectrum of graph product.

Theorem 2.3.30 (Spectrum of graph product, [27])
Let G_1 and G_2 be graphs on n_1 and n_2 vertices, respectively. Then the eigenvalues of $L(G_1 \times G_2)$ are all possible sums $\lambda_i(G_1) + \lambda_j(G_2)$, $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$.

Let us make several consideration about the spectrum of a two dimensional grid graph (Cartesian product of two paths). Consider a product of two paths $P_m \times P_n$, where $m < n$ and their corresponding Laplacian matrices $L(P_m)$, $L(P_n)$ and $L(P_m \times P_n)$. We denote the spectrum $\mathrm{Spec}(P_m) = \{\lambda_1 = 0, \lambda_2, \ldots, \lambda_m\}$, and the spectrum $\mathrm{Spec}(P_n) = \{\mu_1 = 0, \mu_2, \ldots, \mu_n\}$. As $m < n$ then $\lambda_i > \mu_i$, $i < m$ by Theorem 2.3.25.

From Theorem 2.3.30 it simply follows, that the spectrum $\mathrm{Spec}(P_m \times P_n) = \{\lambda_i + \mu_j \mid 1 \leq i \leq m \text{ and } 1 \leq j \leq n\}$. The exact order of the eigenvalues in the spectrum $P_m \times P_n$ depends on the "length" of the paths. Without knowing the exact values m, n, we are not able to distinguish whether $\mu_i + \lambda_j$ is greater then $\mu_k + \lambda_l$ or not ($i, k = 1, \ldots, n$, $j, l = 1, \ldots, m$).

Approximate Cartesian product

For the completeness, the definition of approximate Cartesian product is presented. The definition is adopted from the work of Imrich et al. (see for example [34]) who deal with fast approximate graph product recognition (in polynomial time).

For the definition of approximate graph products we begin with the definition of the distance between two graphs. We say the *distance* $dist(G, H)$ between two graphs G and H is the smallest integer k such that G and H have representations G', H' for which the sum of the symmetric differences ▲ between the vertex sets of the two graphs and between their edge sets is at most k. That is, if

$$|V(G')\blacktriangle V(H')| + |E(G')\blacktriangle E(H')| \leq k.$$

Definition 2.3.31 (Approximate Cartesian product)
A graph G is a k-approximate graph product if there is a product H such that

$$dist(G, H) \leq k.$$

Here k need not be constant, it can be a slowly growing function of $|E(G)|$.

When analyzing the spectrum of approximate Cartesian product, the *behaviour of eigenvectors is similar* to standard Cartesian product (for reasonably small values of k).

2.4 Spectrum of Certain Types of Graphs

In this section, the spectrum of certain types of graphs is presented. As we are interesting in "meshes-like" graphs, the spectrum of paths (corresponding to one-dimensional meshes) and the spectrum of Cartesian products of two paths (corresponding to two-dimensional meshes) is discussed. Similarly, the spectrum of the Cartesian product of three paths (corresponding to three-dimensional meshes) could be investigated, but the eigenvectors of such case are not possible to depict nicely, so this case is omitted.

All investigations about spectrum are made on the Laplacian matrix either with "free" boundary (Neumann boundary condition, standard graph Laplacian matrix according to Definition 2.3.3) or with fixed boundary (Dirichlet boundary condition, according to Equation 2.40).

2.4.1 Spectrum of the path P_n

First, let us have a look on the spectrum of path P_n of length n with vertices $(v_1, v_2, \ldots, v_n)$. Laplacian matrix $L(P_n)$ is a symmetric three-diagonal matrix with twos on the main diagonal except the first and the last elements where ones are, and minus ones on the secondary diagonals. Let us consider Eigenvalue equation (2.1) in the form $L \cdot u = \lambda \cdot u$ as the system of linear equation. In general, we can expect the j-th entry of solution u as a complex number u_j (in this selection, letter i is reserved for notion of the imaginary unit):

$$u_j = e^{i\phi(j+\delta)} = e^{i\phi\delta}e^{i\phi j}. \tag{2.53}$$

For the j-th row, the solution is as follows:

$$\begin{aligned}
-u_{j-1} + 2u_j - u_{j+1} &= \lambda u_j \\
-e^{i\phi\delta}e^{i\phi(j-1)} + 2e^{i\phi\delta}e^{i\phi j} - e^{i\phi\delta}e^{i\phi(j+1)} &= \lambda e^{i\phi\delta}e^{i\phi j} \quad | \cdot \frac{1}{e^{i\phi\delta}} \\
e^{i\phi(j-1)} + 2e^{i\phi j} - e^{i\phi(j+1)} &= \lambda e^{i\phi j} \\
e^{i\phi j}(e^{-i\phi} + 2 - e^{i\phi}) &= \lambda e^{i\phi j} \quad | \cdot \frac{1}{e^{i\phi j}} \\
e^{-i\phi} + 2 - e^{i\phi} &= \lambda.
\end{aligned}$$

From the last equation, λ can be computed as:

$$\begin{aligned}
\lambda &= 2 - 2\frac{e^{i\phi} + e^{-i\phi}}{2} \\
\lambda &= 2 - 2\cos\phi.
\end{aligned}$$

To determine the exact value of the solution u_j (2.53) in each vertex v_j, we have to compute angles ϕ (displacement) and δ (initial phase).

To compute the initial phase δ, we have to use an "extra conditions" to solution u. For this purpose, we expand the path adding two boundary vertices v_0 and v_{n+1}. The conditions corresponding to the Neumann conditions in continuous case (see Subsection 2.1.1) is as follows:

$$u_0 = u_1 \text{ and } u_n = u_{n+1}. \tag{2.54}$$

We can verify, that this condition does not change the behaviour of the solution u on the inner vertices $v_1, v_2, \ldots, v_n$. For $j = 1$, the first row of the system of equations is

$$u_1 - u_2 = \lambda u_1. \tag{2.55}$$

Now, adding vertex v_0, we can write

$$-u_0 + 2u_1 - u_2 = \lambda u_1. \tag{2.56}$$

As we consider condition $u_0 = u_1$, both equations are equivalent. Similarly, condition $u_n = u_{n+1}$ is applied. On the inner vertices $v_1, v_2, \ldots, v_n$ we still obtain the same system of linear equations, i.e. the same solution.

The exact value of δ can be computed from one of the conditions (2.57); let us to evaluate the first one.

$$\begin{aligned} u_0 &= u_1 \\ e^{i\phi\delta} &= e^{i\phi(1+\delta)} \\ \cos(\phi\delta) + i\sin(\phi\delta) &= \cos(\phi(1+\delta)) + i\sin(\phi(1+\delta)). \end{aligned}$$

Comparing the real parts we get

$$\cos(\phi\delta) = \cos(\phi(1+\delta)),$$

which is satisfied either if $\phi\delta = \phi(1+\delta)$ or $\phi\delta = -\phi(1+\delta)$. Choosing the second statement, we get $\delta = -\frac{1}{2}$.

After several (technical) manipulations, we get the exact values for ϕ: $\phi = \frac{t\pi}{n}$, $t = 0, 1, \ldots, n-1$. The corresponding eigenvalues are $\lambda_k = 2 - 2\cos\left(\frac{\pi(k-1)}{n}\right)$, $k = 1, 2, \ldots, n$. As the eigenvalues are real numbers, we consider also real eigenvectors. The j-th entry of the k-th eigenvector is computed as $v_k[j] = \cos\left(\frac{\pi(k-1)}{n}(j - \frac{1}{2})\right)$, $k = 1, 2, \ldots, n$. These results can be found, for example, in [12].

Remark 2.4.1 (Spectrum of $L(P_n)$ "with Neumann boundary")
Consider Laplacian matrix $L(P_n)$ of path P_n on n vertices with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ with corresponding eigenvectors $v_1, v_2, \ldots, v_n$. These are computed as

- $\lambda_k = 2 - 2\cos\left(\frac{\pi(k-1)}{n}\right), \quad k = 1, 2, \ldots, n,$
- $v_k[j] = \cos\left(\frac{\pi(k-1)}{n}(j - \frac{1}{2})\right), \quad k = 1, 2, \ldots, n.$

The $v_k[j]$ denotes the j-th entry of the vector v_k ($j = 1, \ldots, n$).

We can process similarly to obtain the spectrum of Laplacian matrix $\overline{L}(P_n)$ of the path P_n corresponding to the Dirichlet boundary problem (see Subsection 2.3.1). Instead of conditions (2.57) we simply use conditions

$$u_0 = 0 \text{ and } u_{n+1} = 0. \tag{2.57}$$

The resulting Laplacian matrix of inner vertices has twos on the entire main diagonal. Eigenvalues and eigenvectors are computed similarly.

Remark 2.4.2 (Spectrum of $\overline{L}(P_n)$ "with Dirichlet boundary")
Consider Laplacian matrix $\overline{L}(P_n)$ of the path P_n on n vertices with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ with corresponding eigenvectors $v_1, v_2, \ldots, v_n$. These are computed as

- $\lambda_k = 2 - 2\cos\left(\frac{\pi(k-1)}{n}\right), \quad k = 1, 2, \ldots, n,$
- $v_k[j] = \sin\left(\frac{\pi(k-1)j}{n}\right), \quad k = 1, 2, \ldots, n.$

The $v_k[j]$ denotes the j-th entry of the vector v_k ($j = 1, 2, \ldots, n$). Note also, that $\sin(x) = \cos(x - \frac{\pi}{2})$.

Let us have a look on the depiction of the spectrum of a path. In following figures, there are depicted some important eigenvectors corresponding to few of smallest eigenvalues of $L(P_n)$ (Laplacian matrix of graph "with Neumann boundary") and $\overline{L}(P_n)$ (Laplacian matrix of graph "with Dirichlet boundary"), respectively. Eigenvectors corresponding to few of highest eigenvalues of $A(P_n)$ (adjacency matrix) are also depicted.

In Figures 2.1, 2.2(a) the blue coloured lines represent the eigenvectors corresponding to the smallest eigenvalue of $L(P_n)$, $\overline{L}(P_n)$ respectively, the green coloured lines represent the eigenvectors corresponding to the second

smallest eigenvalue, the red coloured lines represent the eigenvectors corresponding to the third smallest eigenvalue, and the cyan coloured lines represent the eigenvectors corresponding to the fourth smallest eigenvalue.

Accordingly, in Figure 2.2(b) the cyan coloured lines represent the eigenvectors corresponding to the highest eigenvalue of $A(P_n)$, the red coloured lines represent the eigenvectors corresponding to the second highest eigenvalue, the green coloured lines represent the eigenvectors corresponding to the third highest eigenvalue and the blue coloured lines represent the eigenvectors corresponding to the fourth highest eigenvalue.

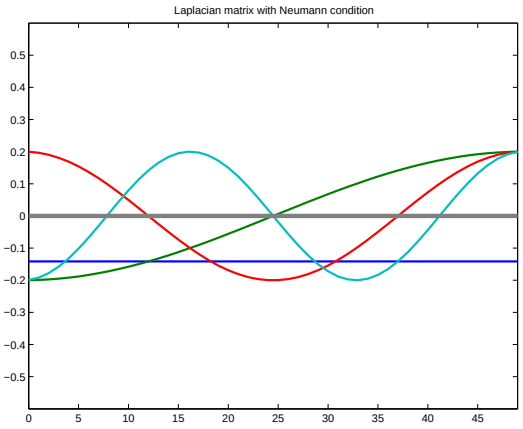
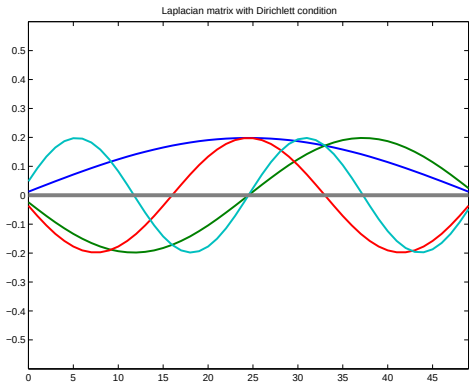
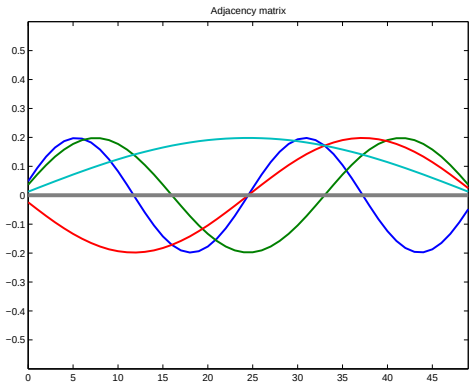


Figure 2.1: Laplacian matrix of path ($L(P_n)$) with Neumann condition



(a) Laplacian m. with Dirichlet cond.



(b) Adjacency m.

Figure 2.2: Laplacian ($\overline{L}(P_n)$) and adjacency matrix ($A(P_n)$) of path

2.4.2 Spectrum of Cartesian product $P_n \times P_m$

To obtain the spectrum of a Cartesian product of two paths we use the knowledge, that the matrices (adjacency and Laplacian, respectively) of Cartesian products are build using Kronecker product. This idea can be found in [12].

Definition 2.4.3 (Kronecker product)
For an $m \times n$ matrix $A = [a_{ij}]$ and $p \times q$ matrix B, the Kronecker product $A \otimes B$ is the block matrix formed by $m \times n$ block of $p \times q$ matrices $a_{ij}B$.

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{2.58}$$

Laplacian matrix L of the Cartesian product is then formed by

$$L(P_n \times P_m) = L(P_n) \otimes I_{P_m} + I_{P_n} \otimes L(P_m), \tag{2.59}$$

where I_{P_n} denotes the unit matrix of the same dimension as $L(P_n)$. Similarly, adjacency matrix A of the Cartesian product is formed by

$$A(P_n \times P_m) = A(P_n) \otimes I_{P_m} + I_{P_n} \otimes A(P_m). \tag{2.60}$$

Based on this knowledge and using some matrix manipulations, we can write the spectrum of Cartesian product of two paths. Here again, standard Laplacian matrix $L(P_n \times P_n)$ corresponds to the graph (mesh) with Neumann boundary.

Remark 2.4.4 (Spectrum of $L(P_n \times P_m)$ "with Neumann boundary")
Consider Laplacian matrix $L(P_n \times P_m)$ of the Cartesian product $P_n \times P_m$ on nm vertices with eigenvalues $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{nm}$ with corresponding eigenvectors $v_{11}, v_{12}, \ldots, v_{nm}$.

- *The eigenvalues are all possible sums of $\lambda_k + \lambda_l$:*
  $\lambda_{kl} = 4 - 2\cos\left(\frac{\pi(k-1)}{n}\right) - 2\cos\left(\frac{\pi(l-1)}{m}\right)$, $\forall k = 1, 2, \ldots, n,\ l = 1, 2, \ldots, m.$
- *The eigenvectors are Kronecker products of eigenvectors v_k, v_l:*
  $v_{kl}[(i-1)m+j] = \cos\left(\frac{\pi(k-1)}{n}(i - \frac{1}{2})\right)\cos\left(\frac{\pi(l-1)}{m}(j - \frac{1}{2})\right)$, $\forall k = 1, 2, \ldots, n$, $l = 1, 2, \ldots, m$.

The $v_{kl}[(i-1)m + j]$ denotes the $((i-1)m + j)$-th entry of the vector v_{kl} $(i = 1, \ldots, n,\ j = 1, \ldots, m)$.

Again, we similarly obtain the spectrum of Laplacian matrix $\overline{L}(P_n \times P_m)$ of the Cartesian product $P_n \times P_m$ corresponding to the Dirichlet boundary problem. Simply, instead of the standard Laplacian matrix $L(P_n)$ and $L(P_m)$, the Dirichlet Laplacian matrices $\overline{L}(P_n)$ and $\overline{L}(P_m)$ are used. Thanks to properties of the Kronecker product, the Dirichlet conditions are distributed from boundaries of paths P_n, P_m to the whole boundary of the Cartesian product $P_n \times P_m$.

Remark 2.4.5 (Spectrum of $\overline{L}(P_n \times P_m)$ "with Dirichlet boundary")
Consider Laplacian matrix $\overline{L}(P_n \times P_m)$ of Cartesian product $P_n \times P_m$ on nm vertices with eigenvalues $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{nm}$ with corresponding eigenvectors $v_{11}, v_{12}, \ldots, v_{nm}$.

- *The eigenvalues are all possible sums of $\lambda_k + \lambda_l$ (i.e. the same as those of $L(P_n \times P_m)$):*
  $\lambda_{kl} = 4 - 2\cos\left(\frac{\pi(k-1)}{n}\right) - 2\cos\left(\frac{\pi(l-1)}{m}\right),\ \forall k = 1, 2, \ldots, n,\ l = 1, 2, \ldots, m.$
- *The eigenvectors are Kronecker products of eigenvectors v_k, v_l:*
  $v_{kl}[(i-1)m + j] = \sin\left(\frac{\pi(k-1)i}{n}\right)\sin\left(\frac{\pi(l-1)j}{m}\right),\ \forall k = 1, 2, \ldots, n,\ l = 1, 2, \ldots, m.$

The $v_{kl}[(i-1)m + j]$ denotes the $((i-1)m + j)$-th entry of the vector v_{kl} $(i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m)$.

Let us have a look on depiction of the spectrum of the Cartesian product. In Figure 2.4 and Figure 2.5, there are depicted couple of important eigenvectors corresponding to the smallest eigenvalues of $L(P_n \times P_m)$ (Laplacian matrix of graph "with Neumann boundary"), $\overline{L}(P_n \times P_m)$ (Laplacian matrix of graph "with Dirichlet boundary") respectively.

As the smallest eigenvalue of $L(P_n \times P_m)$ is zero, the corresponding eigenvector is the constant vector (i.e. each entry has the same value). This case is not depicted. The eigenvector corresponding to the smallest eigenvalue of $\overline{L}(P_n \times P_m)$ is more interesting and it is depicted in Figure 2.3. Let us remark, that this eigenvector forms a "hill" with the top somehow "in the middle of the mesh" (to this consideration is paid attention in Subsection 3.3.2).

In Figure 2.4(a) and Figure 2.4(b), the eigenvectors corresponding to the second smallest eigenvalue of $L(P_n \times P_m)$, $\overline{L}(P_n \times P_m)$ respectively, are depicted. Eigenvectors corresponding to the third smallest eigenvalues are omitted, because they are similar to the eigenvectors corresponding to the second smallest eigenvalues, only the coordinates x, y are switched. To this eigenvectors (second, third and fourth of $L(P_n \times P_m)$) is paid attention in Subsection 3.3.3.

Finally, the eigenvectors corresponding to the fourth smallest eigenvalue of $L(P_n \times P_m)$, $\overline{L}(P_n \times P_m)$ respectively, are depicted in Figure 2.5(a) and Figure 2.5(b).

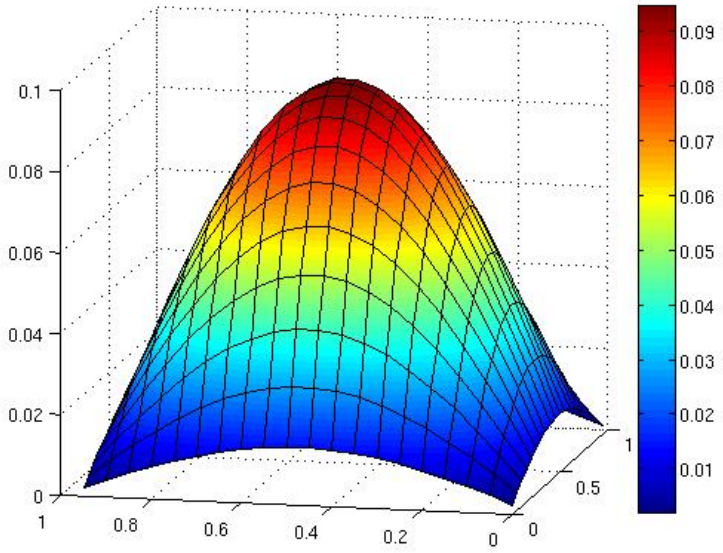
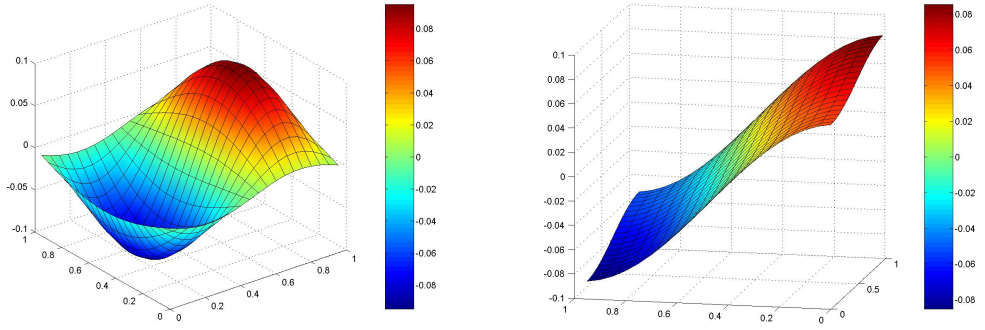
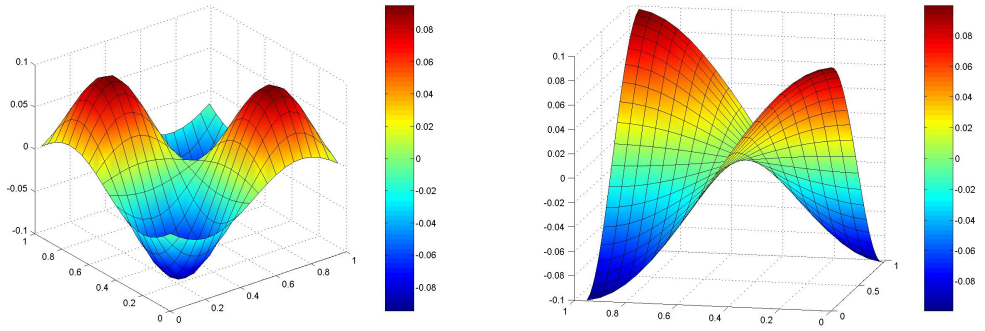


Figure 2.3: $\overline{L}(P_n \times P_m)$, 1st smallest eigenvector



(a) $\overline{L}(P_n \times P_m)$ (b) $L(P_n \times P_m)$

Figure 2.4: Laplacian matrix of Cartesian product, 2nd smallest eigenvector



(a) $\overline{L}(P_n \times P_m)$ (b) $L(P_n \times P_m)$

Figure 2.5: Laplacian matrix of Cartesian product, 4st smallest eigenvector

Chapter 3

Generalized Inverse and Fixing Nodes

3.1 Generalized Inverse

While solving various problems of numerical mathematics, computational mechanics, etc., an SPS matrix often arises. The motivation example from the contact mechanics is described in Section 1.1.

The essential aspect is to solve problem described by Poisson's equation (3.1) or Laplacian's equation (3.2) respectively:

$$\triangle\varphi = f, \tag{3.1}$$

$$\triangle\varphi = 0. \tag{3.2}$$

Depending on the boundary conditions, the resulting discretized problem is described either by an SPD matrix or by an SPS matrix. Hereinafter, the problem with an SPS matrix will be considered.

3.1.1 Neumann Problem

For the sake of simplicity, the system arising from the finite element or finite difference discretization of the two-dimensional Neumann problem shall be concerned:

$$-\Delta u = f \text{ in } \Omega, \qquad \frac{\partial u}{\partial n} = 0 \text{ on } \Gamma, \tag{3.3}$$

where Ω is a domain with a boundary Γ. For meshes with regular elements, the discretization provides exactly the Laplacian matrix $L(G)$ described in

Subsection 2.3.1. In this case, the stiffness matrix arises with the defect equal to one and any diagonal element can be chosen as a zero pivot. From mechanical considerations or from the analysis by Bochev and Lehoucq [7], it is known that the best choice for the position of zero pivot is near the "center" of Ω.

3.1.2 Poisson Equation with Neumann Condition

Finite element solution of problem (3.3) is realized by solving variational formulation of the problem, which is related to minimizing of the energy functional

$$\min_{u \in H(\Omega)} J(u, f), \tag{3.4}$$

where

$$J(u, f) = \frac{1}{2} \int_\Omega |\nabla u|^2 dx - \int_\Omega f u dx \text{ and } f \in L_0^2(\Omega). \tag{3.5}$$

The Euler-Lagrange formula for (3.5) is to seek $u \in H(\Omega)$ such that

$$A(u, v) = b(v) \quad \forall v \in H(\Omega), \tag{3.6}$$

where

$$A(u, v) = \int_\Omega \nabla u \nabla v dx$$

is a bilinear form and

$$b(v) = (f, v)$$

is a linear functional.

The discrete equivalent of (3.5) is

$$\min_u \frac{1}{2} u^T A u - u^T b. \tag{3.7}$$

Solution of (3.7) can be obtained by solving the linear system

$$Au = b$$

with SPS matrix A.

3.1.3 Action of a Generalized Inverse

Let us consider a system of consistent linear equations

$$Kx = b \tag{3.8}$$

with an SPS matrix K. In case of FETI method, matrix K is called the *stiffness matrix*.

If $K \in \mathbb{R}^{n\times n}$ and $b \in \operatorname{Im} K$, where $\operatorname{Im} K$ denotes the range of K, then a solution $x = K^+b$ of the system of linear equations (3.8) can be expressed by means of a *left* (Rao [61], *one-condition*) *generalized inverse matrix* $K^+ \in \mathbb{R}^{n\times m}$, which satisfies

$$KK^+K = K. \tag{3.9}$$

Indeed, if $b \in \operatorname{Im} K$, then there is y such that $b = Ky$ and $x = K^+b$ satisfies

$$Kx = KK^+b = KK^+Ky = Ky = b.$$

Therefore, K^+ acts on the range of K as the inverse matrix. If K is a non-singular square matrix, then obviously

$$K^+ = K^{-1}.$$

3.1.4 Left Generalized Inverse

The Moore–Penrose generalized inverse is known to have the smallest Euclidean norm among all generalized inverse matrices [29], but it is also the most expensive one. On the other hand, to define low expensive inverse of K, observe that if K is a square SPS matrix, then there is a permutation matrix P such that

$$K = P^T \begin{bmatrix} K_{JJ} & K_{JI} \\ K_{IJ} & K_{II} \end{bmatrix} P = P^T \begin{bmatrix} K_{JJ} & C^T \\ C & CK_{JJ}^{-1}C^T \end{bmatrix} P, \tag{3.10}$$

where K_{JJ} is a nonsingular matrix whose dimension is equal to the rank of K. Notice that (3.10) induces the splitting of x and b of (3.8) into x_J, x_I and b_J, b_I respectively. It may be directly verified that the matrix

$$K^+ = P \begin{bmatrix} K_{JJ}^{-1}(K_{JJ} + K_{JI}S_K^\dagger K_{IJ})K_{JJ}^{-1} & -K_{JJ}^{-1}K_{JI}S_K^\dagger \\ -S_K^\dagger K_{IJ}K_{JJ}^{-1} & -S_K^\dagger \end{bmatrix} P^T \tag{3.11}$$

is a left generalized inverse of K such that K^+ satisfies (3.9) and $S_K^\dagger = K_{II} - K_{IJ}K_{JJ}^{-1}K_{JI}$. If the order J of K_{JJ} is equal to the number of degrees

of freedom of given problem, then $S_K^\dagger = O$ (zero matrix of given order) and equation (3.11) can be simplified into

$$K^+ = P \begin{bmatrix} K_{JJ}^{-1} & O^T \\ O & O \end{bmatrix} P^T. \tag{3.12}$$

The goal is to find a permutation matrix P that generates a well conditioned K_{JJ}. The problem is equivalent to the problem of finding the zero pivots that guarantee that the regular part K_{JJ} of K is well conditioned.

More about special types of generalized inverse, about the computation of generalized inverse of an SPS stiffness matrix K, as well as numerous proofs has been written by Alex Markopoulos in his Ph.D. thesis [45].

3.1.5 Another Methods for Computation of the Generalized Inverse Matrix

Several other methods are known how to compute the generalized inverse of an SPS matrix with their advantages and disadvantages. For details see, for example [29].

Singular Value Decomposition

Singular value decomposition (SVD) method produces the Moore–Penrose generalized inverse $A^\dagger$ satisfying the (3.9) condition.

Let us solve the problem $Ax = b$, $A \in \mathbb{R}^{n\times n}$ ($x = A^+b$). The SVD of a matrix A is the decomposition (3.13) and the $A^\dagger$ is then computed by (3.14).

$$A = U\Sigma V^T, \tag{3.13}$$

$$A^\dagger = V\Sigma^\dagger U^T, \tag{3.14}$$

where A is of the defect d, $U, V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, $UU^T = I$, $VV^T = I$, $\Sigma^\dagger = \operatorname{diag}\{\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_{s-d}^{-1}, 0, \ldots, 0\} \in \mathbb{R}^{n\times n}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n-d} > \sigma_{n-d+1} = \cdots = \sigma_n = 0$ are singular values of A.

Cholesky Decomposition

Cholesky decomposition for a regular matrix $Ax = b$, $b \in \operatorname{Im} A$, is the decomposition

$$A = LL^T. \tag{3.15}$$

Suppose that

$$A = \begin{bmatrix} a_{11} & a_1^T \\ a_1 & A_{22} \end{bmatrix} \qquad \text{and} \qquad L = \begin{bmatrix} l_{11} & o \\ l_1 & L_{22} \end{bmatrix}. \tag{3.16}$$

Substituting (3.16) for A and L in (3.15) and comparing the corresponding terms immediately reveals that

$$l_{11} = \sqrt{a_{11}}, \qquad l_1 = l_{11}^{-1} a_1, \qquad L_{22} L_{22}^T = A_{22} - l_1 l_1^T.$$

We can repeat the above procedure until all columns of L are evaluated. The diagonal elements are then computed by (3.17) and the off-diagonal elements by (3.18),

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} a_{ik}^2}, \tag{3.17}$$

$$l_{ij} = \frac{1}{l_{jj}} (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}), \quad i > j. \tag{3.18}$$

The above process is valid for a regular matrix A only. If a matrix $A \in R^{n \times n}$ is an SPS matrix, it can happen that $a_{11} = 0$.

However, the Cholesky decomposition can be adapted to the solution of systems with only positive semi-definite matrix. The only modification comprises in setting to zero the columns that correspond to zero pivots. In agreement with the theoretical results of Pan [53], it turns out that it is very difficult to recognize the positions of such pivots in the presence of rounding errors when the non-singular part of A is ill-conditioned.

Similarly to (3.15), A can be written as

$$A = \begin{bmatrix} 0 & a_1^T \\ a_1 & A_{22} \end{bmatrix} \qquad \text{and} \qquad L = \begin{bmatrix} 0 & o \\ o & L_{22} \end{bmatrix}. \tag{3.19}$$

Substituting (3.19) for A and L in (3.15) and comparing the corresponding terms immediately reveals that

$$l_{11} = 0, \qquad l_1 = 0, \qquad L_{22} L_{22}^T = A_{22}.$$

Again, we can repeat the above procedure similarly as in previous case until all columns of L are evaluated.

Combination of Cholesky Decomposition and SVD

Due to rounding errors, the main difficulty in the implementation of FETI method is an effective elimination of displacements, i.e., an evaluation of the action of the generalized inverse of SPS stiffness matrices of "floating" subdomains. To alleviate this problem, Farhat and Géradin [25] proposed to combine the Cholesky decomposition with the SVD decomposition of a relatively small matrix. The method was developed further by Papadrakakis and Fragakis [54].

Their method consists of two steps. At the beginning, the method starts with the Cholesky decomposition (first step). After revealing the zero pivot, or at least pivot "suspected" to be zero, the method starts with execution of SVD (second step).

The main problem of this method is that the zero pivot can arise at the beginning, which causes the execution of SVD on large matrices and ill-conditioned regular part of the original SPS matrix.

3.2 Fixing Nodes

To solve the problem in the method using combination of the Cholesky decomposition and the SVD presented in the previous Section 3.1 - arising the zero pivot early in the algorithm and ill-conditioning of the regular part of the SPS matrix A, a new technique has been developed. An improved modification of this method was proposed by Brzobohatý, Dostál, Kozubek, Kovář, and A. Markopoulos in [9]. This modification, based on an active choice of the SVD part, uses "fixing nodes" strategy to make the system as stiff as possible and it has been implemented in the MatSol library [46], as a part of the Total–FETI algorithm.

The active choice is made in sense of permutation matrix P in equation (3.12), i.e., the rows/columns corresponding to those vertices marked as so-called "fixing nodes" are permuted at the end of the original stiffness matrix. Also, the rigid body motions corresponding to degrees of freedom of selected "fixing nodes" are removed.

The minimization of functional (3.20) is considered,

$$J(v) = \min \frac{1}{2} v^T K v - v^T f, \tag{3.20}$$

together with condition

$$Bv - g = 0, \tag{3.21}$$

where matrix B enforces the removing of the rigid body motions. Minimum of this functional is obtained in sense of Lagrange multipliers.

It is clear, that the number of "fixing nodes" is quite small, as it corresponds to the number of degrees of freedom. In case of two-dimensional linear elastic structure, the minimum number of "fixing nodes" is two to prevent all rigid body motions (translation and rotation). In case of three-dimensional problem, the minimum number of "fixing nodes" is three. Those three "fixing nodes" should not be placed near any straight line. When we increase the number of "fixing nodes", the condition number of regular part K_{JJ} decreases, but the computational cost of the SVD of singular part increases. Also, finding more "fixing nodes" takes more computation time. So, it is necessary to find a suitable compromise. In our tests, it turns out that it is sufficient to take four "fixing nodes" to get reasonably small condition number.

At the beginning of this section, I present the explicit definitions of fixing nodes. Finding fixing nodes effectively and accurately is the main task of this section. Because Definition 3.2.1 as well as Definition 3.2.2 operate on condition numbers of residual matrices, they are not suitable to evaluate the

fixing nodes in reasonable time. Therefore, a fast heuristic has been looked for to provide a good approximation of the fixing node. In this section, I propose several approaches how to find a good approximation of one fixing node that can be extended to finding approximation of more fixing nodes. The results of "positioning" of fixing nodes based on listed approaches are presented at the end of this section. In Section 4.1, it is shown that one of these approaches leads to the best choice of one–fixing node.

For purposes of this section, the notation established in Section 2.3 is used. The problem (3.3) is considered in the discretized form (3.7) with the corresponding linear structure (3.8). The discretization of the problem corresponds to given numerical mesh. This mesh can be considered as unweighted, non-oriented graph and matrices of this graph (adjacency, Laplacian) can be generated. Discrete Laplacian matrix $L(G)$ differs from stiffness matrix K considered in Subsection 3.1.3 and 3.1.4 only by a scalar factor, thus the spectral properties of both matrices are the same.

3.2.1 Definitions of Fixing Nodes

Consider an SPS stiffness matrix K. As the matrix K is semi-definite, its smallest eigenvalue λ_{min} is equal to zero. The condition number of K computed as $\lambda_{max}/\lambda_{min}$ is therefore equal to infinity (from numerical point of view).

Recall the decomposition of the stiffness matrix K from Subsection 3.1.4:

$$K = P^T \begin{bmatrix} K_{JJ} & K_{JI} \\ K_{IJ} & K_{II} \end{bmatrix} P, \tag{3.22}$$

where K_{JJ} is a non-singular matrix which dimension is equal to the rank of K. The process of finding fixing nodes then consists of finding permutation matrix P, such that the rows/columns corresponding to singular part K_{II} are permuted to the bottom-right block of the matrix.

From the considerations above, the following definition of one fixing node can be composed.

Definition 3.2.1 (one–fixing node)
Let $Kx = b$ be a system of linear equations arising from a finite element or finite difference discretization of the problem (3.3), such that K has one-dimensional kernel (i.e., the singular part K_{II} in (3.22) is formed by one zero element).

The one–fixing node[1] *is the node that makes the regular part K_{JJ} of the*

[1] In further text, we will use the term "fixing node" instead of the term "one–fixing node" when no misunderstanding could happen.

stiffness matrix K produced by the permutation P non-singular and well conditioned, i.e., permutation of this node to the last row/column of the matrix K makes the condition number of the regular part K_{JJ} finite and sufficiently small (see Equation (3.22)).

The best choice of one–fixing node *is the node k for which the regular part K_{JJ} (assigned as $\widetilde{K}$)[2] of the stiffness matrix K produced by the permutation P has the minimal condition number over all $\widetilde{K}_{\widehat{kk}}$,*

$$\mathrm{cond}(\widetilde{K}) = \min_{k=1,\ldots,n} \mathrm{cond}(\widetilde{K}_{\widehat{kk}}).$$

Condition number $\mathrm{cond}(\widetilde{K}_{\widehat{kk}})$ is computed as

$$\mathrm{cond}(\widetilde{K}_{\widehat{kk}}) = \frac{\lambda_{max}(\widetilde{K}_{\widehat{kk}})}{\lambda_{min}(\widetilde{K}_{\widehat{kk}})},$$

where $\lambda_{max}(\widetilde{K}_{\widehat{kk}})$ and $\lambda_{min}(\widetilde{K}_{\widehat{kk}})$ denote the largest and the smallest eigenvalue of the regular matrix $\widetilde{K}$, respectively.

In case of two-dimensional problem, one node corresponds to two rows and columns in the matrix K of given problem. In case of three-dimensional problem, one node corresponds to three rows and columns in the matrix K of given problem. Rearrangement of such node consists in rearrangement of all corresponding rows and columns.

Extending these considerations, definition of more fixing nodes can be composed accordingly:

Definition 3.2.2 (i-fixing nodes)
The i-fixing nodes[3] *are the set of i nodes that make the regular part K_{JJ} of the stiffness matrix K produced by the permutation P nonsingular and well conditioned, i.e., permutation of these nodes to the bottommost rows, rightmost columns respectively, of the matrix K makes the condition number of regular part K_{JJ} finite and sufficiently small (see Equation (3.22)).*

The best choice of i-fixing nodes *is the set of i-fixing nodes for which the regular part K_{JJ} of the stiffness matrix K produced by the permutation P has the minimal condition number over all such permutations.*

[2]Note, that the K_{JJ} is the principal submatrix of K. As the symbol K_{JJ} denote the submatrix of matrix K of order J, for the principal submatrix arising by removing k-th row and column we have to use notation $K_{\widehat{kk}}$. For the problems with one-dimensional kernel, the $J = n - 1$ and we denote the matrix K_{JJ} as $\widetilde{K}$ to avoid misunderstanding.

[3]If the number i of i-fixing nodes is not important, we use only the term "fixing nodes" instead of the term "i-fixing nodes".

Further in this text, I show techniques how to find one–fixing node. One–fixing node can stabilize well the solution of problems with one-dimensional kernel, like (3.3).

When computing with another types of problems, or when the problem is large, it may be necessary to modify the procedure so that it works for several fixing nodes. Such problem can be reduced to the problem of finding one-fixing node by decomposition of the mesh by a suitable software into several parts as it is illustrated in Figure 3.1. The fixing nodes are then found one in each part.

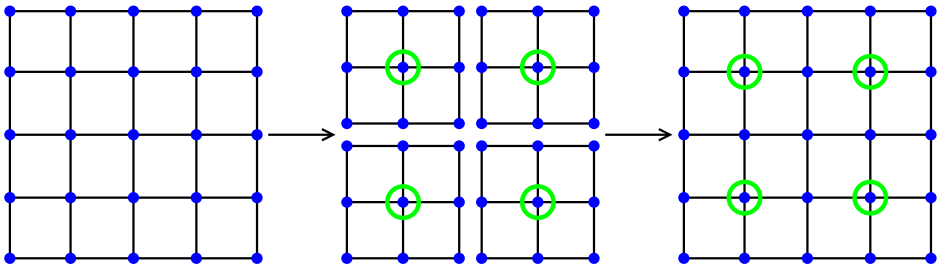


Figure 3.1: Four fixing nodes

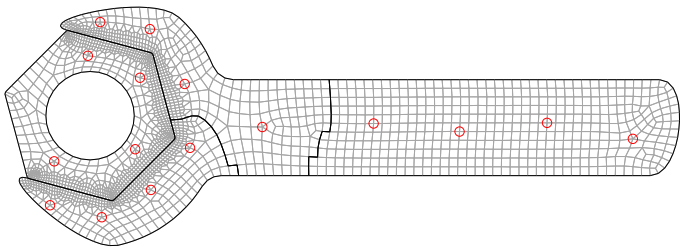


Figure 3.2: Four fixing nodes in four domains

In this case, it is guaranteed that we can obtain the best choice of one–fixing node in each part in sense of Definition 3.2.1 but the resulting set of i-fixing nodes of the whole problem do not need to be the best choice of i-fixing nodes in sense of Definition 3.2.2. To get as good position as possible (in sense of minimizing the condition number of the regular part of the matrix), it is strictly required that the resulting subdomains fulfil the requirements on the optimal decomposition presented in Section 1.2. Again, Metis software [47] is used with some post-processing.

When the problem is decomposed into several floating subdomains, the

desired number of fixing nodes is found in each subdomain. Such example is depicted in Figure 3.2.

Various tests corresponding to positioning of fixing nodes were made in [9, 42, 45]. From mechanical point of view, it appears to place fixing nodes uniformly in the mesh. Considering the mesh to be the non-oriented graph, one can consider the problem of finding fixing nodes as the problem of finding "graph centers" (in certain case). This idea is described in [42].

3.3 Center-like Points of Graphs

As the notation of "graph center" or "graph centers" respectively evokes some of various types of standard graph definition, I rather propose to call these points *center-like points* of graph as they lie somehow near the geometrical center of graph (in case of regular mesh with constant edge-length).

I would like to show that some candidates deliver a good approximation of the best choice of one–fixing node. First of all, let us show that one of standard definitions of graph center can fulfil our requirements, second, some other candidates based on spectral graph theory will be also shown.

Finding graph centers has huge computational complexity and it is not suitable for the numerical solution of large problems. In practical parallel computation of large scale problems, the fast computation is preferred even if the result sometimes differs from the best position of fixing nodes. Therefore, a fast heuristic has been looked for to provide a good approximation of the best choice of one–fixing node. Here, I examine and compare three such methods: using a center of graph, using Perron vector (an eigenvector corresponding to the highest eigenvalue) of the adjacency matrix, and using eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix.

3.3.1 Graph Center

One of the approaches to find an approximation of the best choice of one–fixing node is based on the mechanical interpretation of the problem. It follows the idea to choose the fixing node near the "center" of the mesh. This translates directly into choosing one of the vertices in the center of the graph. Finding graph centers is not suitable for the numerical solution of large problems but it provides a good referential point to the other methods. The results of this subsection has been published in [42].

As we are interested only in the structure of the mesh, adjacency matrix $A(G)$ of corresponding graph G is used to graph center finding. The graph center is the set of vertices, in which the minimum eccentricity is realized (see Definition 2.2.4). The definition of graph center for one vertex follows:

Definition 3.3.1 (Graph center)
Let the graph center *of a graph G be a vertex x that satisfies*

$$\min_{x \in V(G)} \max_{v \in V(G)} \operatorname{dist}(x, v), \tag{3.23}$$

where $V(G)$ is the vertex set of graph G, $\operatorname{dist}(x, v)$ *is the distance between vertices x and v.*

There are several ways to generalize the notion of the graph center to more vertices. One is the following.

Definition 3.3.2 (Graph centers)
Let the graph k-center *of a graph G be a set C of k vertices for which*

$$\min_{\substack{C\subset V(G)\\|C|=k}} \max_{v\in V(G)} \operatorname{dist}(C,v) = \min_{\substack{C\subset V(G)\\|C|=k}} \max_{v\in V(G)} \left(\min_{x\in C} \operatorname{dist}(x,v) \right), \tag{3.24}$$

where k is the number of vertices in the center C, $V(G)$ is the vertex set of graph G, $\operatorname{dist}(x,v)$ *is the distance between vertices x and v (length of the shortest path between these vertices).*

Finding graph center based on Definition 3.3.1 is of the complexity $O(n^3)$, thus it is not possible to use this definition for large-scale meshes[4]. Finding graph k-center based on Definition 3.3.2 is of the polynomial complexity, which is not suitable even for small meshes. If one wish to find an approximation of the k-fixing nodes based on the graph centers, it is better idea to decompose given mesh into k domains and find one graph center (Definition 3.3.1) in each part, see paragraphs below Definition 3.2.2.

Finding graph center based on Definition 3.3.1 can yield several adepts for an approximation of fixing node, since the graph center can contain arbitrary many vertices. Though, in graphs corresponding to square meshes, there will not be many of them (see the figures in Subsection 3.3.4). We can choose one adept closest to the geometrical center. This chosen vertex does not always fulfil Definition 3.2.1 (best choice of one–fixing node) but it is sufficiently close to the desired node.

3.3.2 Perron Vector of the Adjacency Matrix

Another idea how to identify the fixing node is based on spectral approach, namely on eigenvector of the adjacency matrix of graph. Again, adjacency matrix $A(G)$ of corresponding graph G is used.

The following lemma is well known.

Lemma 3.3.3 (k-th power of A)
Let A be the adjacency matrix of a given mesh and let $B = A^k$. Each element b_{ij} of B gives the number of distinct (i,j)-walks[5] of length k in the mesh.

[4]Distances $\operatorname{dist}(u,v)$ form a dense $n \times n$ matrix even for sparse matrices A.
[5]The walk is defined in Definition 2.2.2.

The eigenvector that corresponds to the largest eigenvalue of matrix A is referred as the *Perron vector*. It could be numerically computed for example by the, so-called, Power method ($w_k = A^k w_0$, where initial w_0 is the constant vector $(1, 1, \ldots, 1)$), see for example [29]. The definition of the *capital vertex* follows.

Definition 3.3.4 (Capital vertex)
Let the capital vertex *of a graph G be a vertex v_i that corresponds to the index i of the highest value in the Perron vector (the eigenvector that corresponds to the largest eigenvalue) of the corresponding adjacency matrix $A(G)$ of graph G.*

Notice, that this numerical computation together with Lemma 3.3.3 suggests that the largest coordinate i of the Perron vector represents the total number of walks of length k starting at vertex v_i. Therefore, the capital vertex does not identify the center of the graph, rather a vertex from which most walks of a given length can be realized. Such a vertex is assumed to have the most influence on rounding error distribution during the numerical computation on the mesh.

By choosing the capital vertex to be the fixing node we expect to achieve a stable numerical solution. Moreover, identifying the capital vertex is fast. Finding the approximation of the Perron vector has complexity roughly $O(kn^2)$ (k iterations of matrix times vector). For sparse matrices, the complexity $O(kn)$ can be expected, where k is considered as a parameter of the numerical approximation. Already for small values of k (units or dozens), this gives a good approximation of the Perron vector, while n is in hundreds of thousands. The parameter k is usually chosen in hundreds even for large-scale meshes.

3.3.3 Eigenvectors of the Laplacian Matrix

The last approach how to obtain a good approximation of the best choice of one–fixing node mentioned in this text is based on the eigenvectors of the Laplacian matrix. Let us discuss two main reasons why to use the Laplacian matrix instead of the adjacency matrix of the mesh.

First, the adjacency matrix considers only the mesh/graph structure, i.e., between which nodes the edge is, the diagonal entries are empty. The row's/column's sums differ in dependence on degree of the given vertex. The Laplacian matrix takes care also on the degrees of vertices - counted in the diagonal elements. Moreover, the standard Laplacian matrix has all row's/column's sums equal to zero. The Laplacian matrix can be also set up

in dependence on the boundary conditions. As standard Laplacian matrix (see Subsection 2.3.1) corresponds to problem with the Neumann boundary (see Subsection 3.1.1), it is more suitable to represent the problem that we are interesting in.

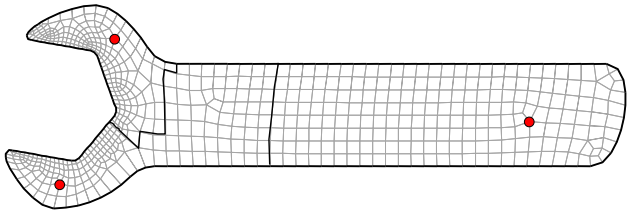


Figure 3.3: Mesh with wrong position of fixing nodes

Second, in results of authors Brzobohatý, Dostál, Kozubek, Kovář, and Markopoulos in [9] (see Figure 3.3 taken from this paper), the position of the fixing node on the rightmost side subdomain differs from the position where we naturally expect it should be (i.e. closer to the "center" of given subdomain). In [9], authors have used the approach based on the Perron vector of the adjacency matrix. One can see that the highest value of the Perron vector arises at the vertex with the higher degree rather than in one with the mean value of degree. The explanation of this phenomenon is given later in this text.

Before presenting a new definition of an another center-like vertex, let us recall several known results from spectral graph theory (see Section 2.3). According to M. Fiedler theory [27], [28], the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix has been used to graph partitioning [4], [40], [59]. It is known, that the so-called *Fiedler cut* defines the cut in certain coordinate direction. The interesting thing is that in two-dimensional case, the eigenvector corresponding to the third smallest eigenvalue defines the cut in the second coordinate direction. Because the Fiedler cut is known to be somehow "optimal" (not in the sense of optimal decomposition defined in the Subsection 1.2.1), it is reasonable to expect a good approximation of one–fixing node near the crossing of both cuts.

For purposes of definition of a generalized eigenvector cut, the notation based on "Fiedler's tree theorem" (Theorem 2.3.18) is used, especially notation of the *characteristic vertex* and the *characteristic edge*.

Using other Fiedler's results (some of them are listed in Subsection 2.3.3), the *Fiedler cut* is produced by the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix, and it is formed by the *character-*

istic set (i.e. set of characteristic vertices or characteristic edges). In general, *characteristic set* can be defined for arbitrary eigenvector of the Laplacian matrix (as in Theorem 2.3.18) and for arbitrary type of graph.

Definition 3.3.5 (k-level cut)
Let $G = (V, E)$ be a graph on n vertices, labelled $1, 2, \ldots, n$, with Laplacian matrix $L(G)$. Let v_k be an eigenvector of $L(G)$ associated with the k-th eigenvalue.

The k-level cut is defined as a characteristic set, i.e. either set of characteristic vertices or set of characteristic edges, where

- *characteristic vertices are the vertices i that satisfy $v_k[i] = 0$.*
- *characteristic edges are the edges ij such that i and j are adjacent in G, $v_k[i] > 0$ and $v_k[j] < 0$ (or $v_k[i] < 0$, $v_k[j] > 0$).*

Following this notation, the 2-level cut is exactly the Fiedler cut. Using notation of Theorem 2.3.23 (Discrete Nodal Domain Theorem), we can remind that the eigenvector v_k corresponding to the k-th eigenvalue with multiplicity r divides graph into at most k weak nodal domains (with zero-valued elements) and $k + r - 1$ strong nodal domains (without zero-valued elements).

Now, the *cross-eigenvector center* can be defined. As the spectrum, i.e. the vibration model (see Subsection 2.1.1), respects the physical properties of given objects, i.e. the space-dimension, also the cross-eigenvector center has to be defined with respect to this characteristic (dimension of the mesh).

Definition 3.3.6 (Cross-eigenvector center)
Let $G = (V, E)$ be a graph on n vertices.

- *For one-dimensional mesh, the* cross-eigenvector center *is the vertex or edge respectively that lies on the 2-level cut*[6].
- *For two-dimensional mesh, the* cross-eigenvector center *is the vertex, edge or two-dimensional element respectively that lies on the crossing of the 2-level cut and the 3-level cut.*
- *For three-dimensional mesh, the* cross-eigenvector center *is the vertex, edge, face or three–dimensional element respectively that lies on the crossing of the 2-level cut, 3-level cut and 4-level cut.*

[6]Remark that the one-dimensional mesh is a tree, i.e. the cross-eigenvector center is defined exactly as the characteristic vertex or characteristic edge according to Theorem 2.3.18.

There are three possibilities, where the cross-eigenvector center can occur:

1. If all of the corresponding k-level cuts consist of the characteristic vertices, **the cross-eigenvector center is the vertex**.

2. If one of the corresponding k-level cuts in two-dimensional case consists of the characteristic edges, or two in three-dimensional case respectively, **the cross-eigenvector center is the edge**.

3. If all of the corresponding k-level cuts case consist of the characteristic edges, or at least two in three-dimensional case respectively, **the cross-eigenvector center is an element** (or face respectively)[7].

In computational arithmetic, i.e. in presence of rounding errors, there is usually problem with zero identification. The cross-eigenvector center is always assigned to the vertex with the smallest value (in absolute value). The exact position of the cross-eigenvector center can differ no more than half edge, half element respectively. Also, as the other center-like points are defined in the vertex, it is reasonable to consider the cross-eigenvector center as the vertex too.

Further in this text, we would like to prove that the cross-eigenvector center delivers the best choice of one–fixing node (instead of the other center-like points), which is its biggest advantage. In opposite to the capital vertex, the eigenvectors corresponding to the second, third, and fourth smallest eigenvalue of the Laplacian matrix are harder to compute than the eigenvector corresponding to the largest eigenvalue of the adjacency matrix, which is a disadvantage.

The research of the fast algorithm to compute these eigenvectors of Laplacian matrix even for large-scale and bad conditioned problems is left for further work.

3.3.4 Approximation of Fixing Node by Center-like Points

In this section, the results of finding a good approximation of the best choice of one–fixing node are presented on a given testing set. The criterion to measure quality of the approximation of one–fixing node is the condition number of the regular part K_{JJ} of the matrix K of given problem ((3.12) or (3.22) respectively) after permutation of the row and column corresponding to the fixing node at the end of the matrix.

[7]The element is defined as a part of the plane bounded by edges (i.e. face), or in three-dimensional space as a space bounded by two-dimensional faces.

The testing has been done on a given set of examples and the resulting figures have been generated. Given examples are quite small (less than 121 vertices and 100 edges) because the finding of the best choice of one–fixing node according to Definition 3.2.1 yields to removing all vertices, computing the condition number of the regular part of the matrix (i.e. whole spectrum) for each removal, and choosing the minimal one, which is a huge time-consuming process.

In figures the following symbols appear:

1. The *best choice of one–fixing node* satisfying Definition 3.2.1 is drawn as a circle ◯.
2. The *graph center* satisfying Definition 3.3.1 is drawn as a square ■. When more vertices satisfy the definition, all such vertices are drawn.
3. The *capital vertex* satisfying Definition 3.3.4 is drawn as a triangle △.
4. The *cross-eigenvector center* satisfying Definition 3.3.6 is drawn as a star ✳.

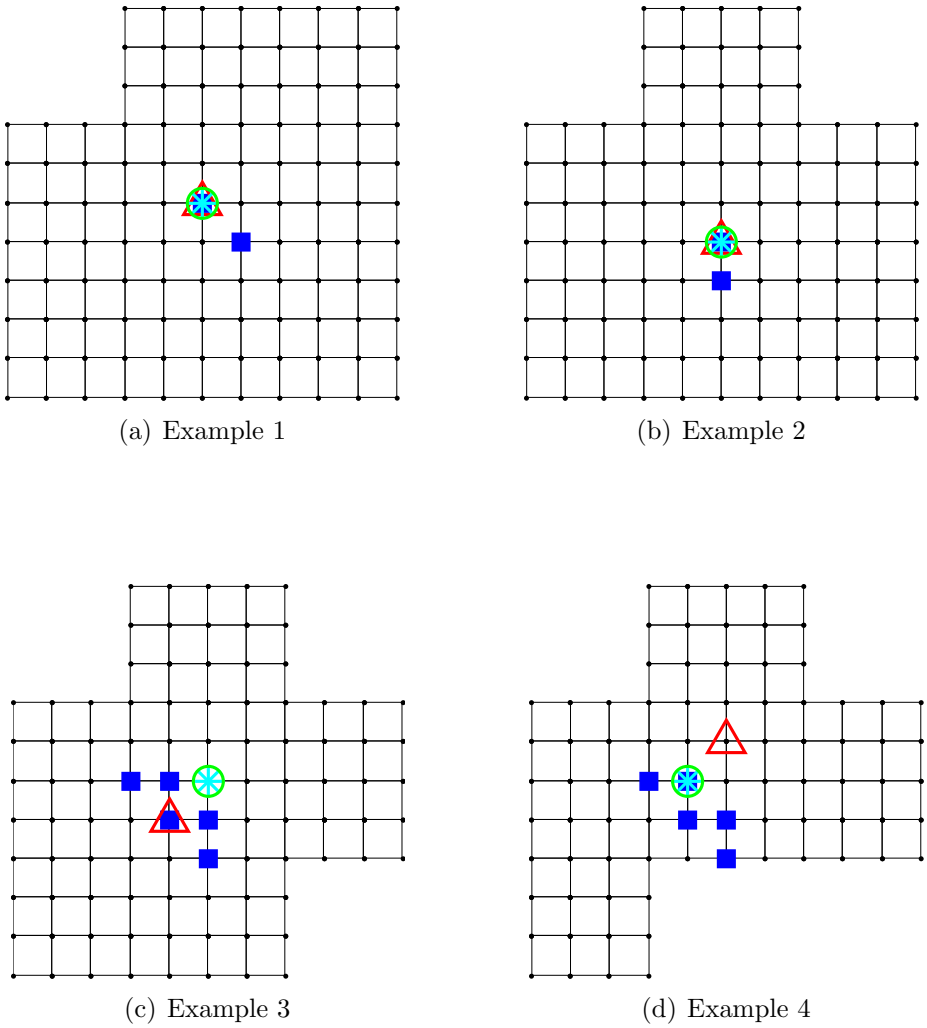
Meshes with uniform elements

At first, let us have a look on the behaviour of center-like points on the meshes with regular elements. In two-dimensional space, the element corresponds to the graph face (region bounded by edges). Our examples consist of quadrilateral type elements only.

As we can see in Figure 3.4(a)–(d), all these vertices are somehow in the "center" of the mesh. Moreover, the more compact the given mesh is, the closer these vertices are. Table 3.1 shows condition numbers related to the testing examples.

In Table 3.1, the first column represents the name of the test example. The regular condition number of the matrix K in the second column is computed by $\overline{\mathrm{cond}}(K) = \frac{\lambda_{max}(K)}{\overline{\lambda}_{min}(K)}$, where $\lambda_{max}(K)$ and $\overline{\lambda}_{min}(K)$ correspond to the largest and to the smallest non-zero eigenvalue of K. The regular condition number is used as a reference value, because it represents the smallest boundary to the condition number of the generalized inverse (the condition number of the generalized inverse can be never smaller than the condition number of the original matrix). The third column represents the condition number of the matrix K_{JJ} considered in Definition 3.2.1, i.e. the minimal possible condition number of the regular part K_{JJ} after removing the best choice of one–fixing node. The condition number of the regular part K_{JJ} after removing the graph center satisfying Definition 3.3.1 is written in the fourth column. If more vertices satisfy the definition, the vertex that causes the

minimal condition number has been chosen. The fifth column represents the condition number of the matrix K_{JJ} when the capital vertex satisfying Definition 3.3.4 is removed. The last column represents the condition number of the matrix K_{JJ} when the cross-eigenvector center satisfying Definition 3.3.6 is removed.



(a) Example 1 (b) Example 2

(c) Example 3 (d) Example 4

Figure 3.4: Approximation of the best choice of one–fixing node

As we can see from the figures and from the table, almost all approaches deliver a good approximation of one–fixing node. In practical parallel computation of large scale problems, the fast computation needed for the capital vertex is preferred even if it sometimes differs from the best choice of one–

fixing node. The approach using the capital vertex is already implemented in the MatSol library [46] and it has been tested on large scale problems [21].

No.	$\overline{\text{cond}}(K)$ (regular)	$\text{cond}(\widetilde{K}_{JJ})$ (1-fixing node)	$\text{cond}(\widetilde{K}_{JJ})$ (graph center)	$\text{cond}(\widetilde{K}_{JJ})$ (adjacency m.)	$\text{cond}(\widetilde{K}_{JJ})$ (Laplace m.)
1	107.7	379.8	379.8	379.8	379.8
2	90.8	341.1	341.1	341.1	341.1
3	98.4	308.8	311.8	317.0	308.8
4	126.4	273.1	273.1	289.5	273.1

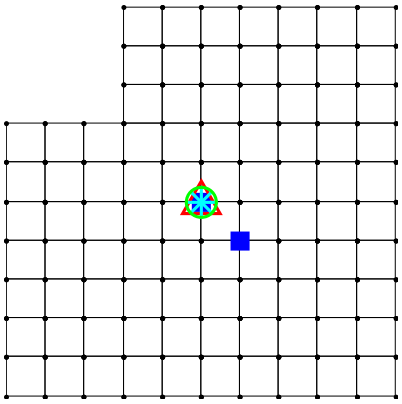
Table 3.1: Approximation of the best choice of one–fixing node (1)

Meshes with non-uniform elements

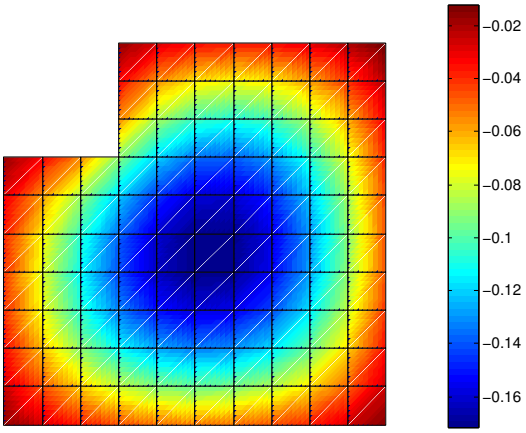
From the previous testing examples, one can suppose that the behaviour of all center-like points is similar. But this holds only for meshes with uniform elements, or similarly, for meshes with inner vertices of the same degree. I have chosen the mesh with mixed elements (quadrilateral and triangle type) for further testing. This means that the degree of inner vertices differ from four to six.

Let us have a look in Figures 3.5, 3.6. In Figure 3.5(a), Example 1 from the previous test is shown. Moreover, the eigenvectors are plotted that we are interesting in (basic concept about the spectrum of meshes-like graphs is provided in Section 2.4). In Figure 3.5(b), the eigenvector of the adjacency matrix corresponding to the highest eigenvalue is plotted. In Figure 3.5(c), the eigenvector of the Laplacian matrix corresponding to the second smallest eigenvalue is plotted, and in Figure 3.5(d), the eigenvector of the Laplacian matrix corresponding to the third smallest eigenvalue is plotted. There is nothing interesting in these figures. Our mesh (Example 1) is similar to the Cartesian product of two paths $P_{11} \times P_{11}$ and also, the spectrum of our mesh is similar to the spectrum of Cartesian product $P_{11} \times P_{11}$. The mesh Example 1 (as well as Example 2-4) can be understood as an approximate Cartesian product with deleted both edges and vertices. The missing left upper corner has no essential influence on the behaviour of the eigenvectors.

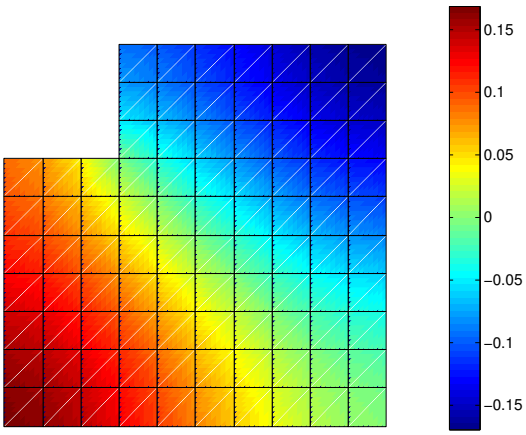
Remark 3.3.7 (Spectrum of approximate Cartesian product)
Consider the approximate graph product satisfying Definition 2.3.31. The behaviour of the eigenvectors of the approximate Cartesian product types $P_n \times P_n$ is similar to the original Cartesian products.



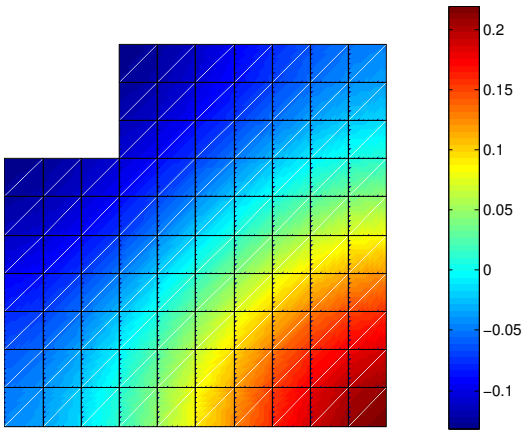
(a) Example 1



(b) Adjacency m., first eigenvector



(c) Laplace m., second eigenvector



(d) Laplace m., third eigenvector

Figure 3.5: Example 1 with plotted eigenvectors

Remark 3.3.8 (Center-like points of approximate Cartesian product (removing edges/vertices))
As the behaviour of the eigenvectors of the approximate Cartesian product types $P_n \times P_n$ is similar to the original Cartesian products, all types of center-like points *of the approximate Cartesian product approximate the best choice of one–fixing node well when reasonably small number of edges/vertices is removed from corresponding original Cartesian product.*

Another situation arises if the edges are added instead of deleted. This situation is depicted in Figure 3.6(a). Again, In Figure 3.6(b), the eigenvector of the adjacency matrix corresponding to the highest eigenvalue is plotted. In Figure 3.6(c), the eigenvector of the Laplacian matrix corresponding to the second smallest eigenvalue is plotted, and in Figure 3.6(d), the eigenvector of the Laplacian matrix corresponding to the third smallest eigenvalue is plotted.

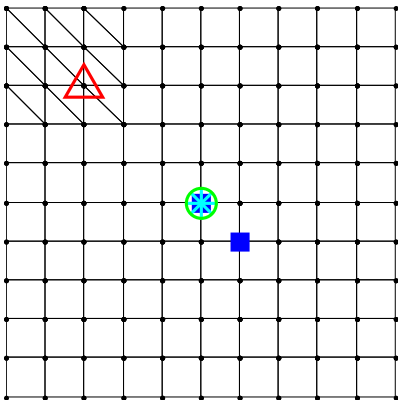
As we can see, the behaviour of the eigenvector of the adjacency matrix corresponding to the highest eigenvalue does not correspond to the classical behaviour of the spectrum of Cartesian products. Therefore, the capital vertex (based on the adjacency matrix) do not approximate the best choice of one–fixing node well.

As we can see in Figures 3.6(c), 3.5(d), the eigenvectors of the Laplacian matrix are still indifferent to changing of the structure of the mesh. The cross-eigenvector center still approximates the best choice of one–fixing node well.

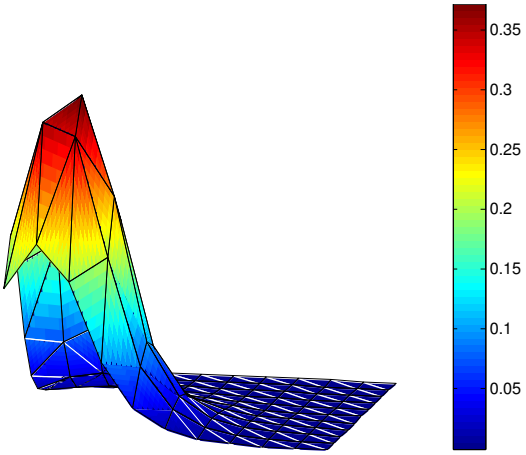
For better comprehension, Table 3.2 is presented. The notion is the same as in previous Table 3.1. We can see that choosing the capital vertex as the approximation of the best choice of one–fixing node instead the cross–eigenvector center debases the size of the condition number from 426.5 (best choice of one–fixing node, cross-eigenvector center) to 537.1, i.e., by 26%.

No.	$\overline{\text{cond}}(A)$ (regular)	$\kappa(\widetilde{A}_{JJ})$ (1-fixing node)	$\text{cond}(\widetilde{A}_{JJ})$ (graph center)	$\text{cond}(\widetilde{A}_{JJ})$ (Adjacency m.)	$\text{cond}(\widetilde{A}_{JJ})$ (Laplace m.)
1	107.7	379.8	379.8	379.8	379.8
5	99.2	426.5	426.5	537.1	426.5

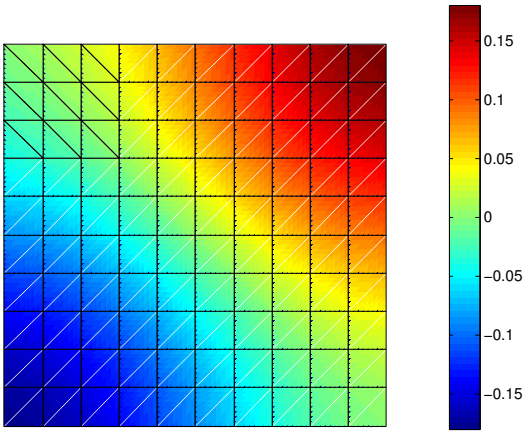
Table 3.2: Approximation of the best choice of one–fixing node (2)



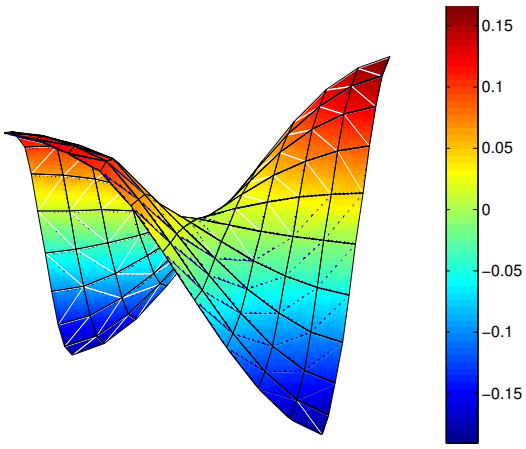
(a) Example 5



(b) Adjacency m., first eigenvector



(c) Laplace m., second eigenvector



(d) Laplace m., fourth eigenvector

Figure 3.6: Example 5 with plotted eigenvectors

Remark 3.3.9 (Center-like points of approximate Cartesian product (adding edges))
For meshes like approximate Cartesian product with additional edges, the behaviour of eigenvectors of the adjacency matrix differs substantially from the original Cartesian product and the capital vertex *is not suitable approximation for these types of graphs.*

The best approximation of the best choice of one–fixing node is then the cross-eigenvector center.

If we consider the meshes (graphs respectively) corresponding to numerical problems, the first case (missing edges/vertices) is more frequent. The eigenvector of the adjacency matrix works well and the capital vertex is a good approximation of the best choice of one–fixing node.

However, the elements in the numerical mesh can "degenerate", i.e., a quadrilateral element can degenerate into a triangle element (it depends on a shape complexity of numerical object, on a quality of the mesh generator, etc.), which causes the second case with additional edges. Here, the eigenvector of the adjacency matrix does not work well and the capital vertex is not a good approximation of the best choice of one–fixing node. Thus, the eigenvectors of the Laplacian matrix have to be used.

Chapter 4

The Best Choice of One–Fixing Node

4.1 The Best Choice of One–Fixing Node

In Section 3.2, various approaches have been used to find a good approximation of the best choice of one–fixing node (see Definition 3.2.1). Based on the experiments, Remark 3.3.9 claims that the best approximation of the best choice of one–fixing node is the *cross-eigenvector center*. In this section, I would like to prove this claim analytically.

As it is known from the definition of fixing node (Definition 3.2.1), the quality of the approximation of the best choice of one–fixing node is measured by the condition number of the stiffness matrix of a corresponding problem. As it has been noted, the stiffness matrix has the same structure as the graph Laplacian matrix of corresponding mesh. Therefore, the Laplacian matrix is used for the proof.

I would like to prove following theorem:

Theorem 4.1.1 (The best choice of one–fixing node)
The cross-eigenvector center (Definition 3.3.6) is the best choice of one–fixing node (Definition 3.2.1).

In other words, if the row and column of the original matrix K that correspond to the cross-eigenvector center of corresponding mesh are removed, the remaining principal submatrix has the the minimal condition number over all principal submatrices[1].

[1] In practical computing, the process of "fixing some vertex" consists in prescription of the Dirichlet condition to given vertex.

Fixing one node means deleting one (corresponding) row and column from the Laplacian matrix $L(G)$, i.e., the resulting matrix is the principal submatrix of $L(G)$, denoted as $\widetilde{L}(G)$. Note, that the $\widetilde{L}(G)$ is no longer the Laplacian matrix in sense of Definition 2.3.3 and it has no longer the smallest eigenvalue equal to zero!

Further in this section, all variables corresponding to the reduced problem will be written with $\sim$, e.g., $\widetilde{L}$, $\widetilde{\lambda}$, etc. If we could to emphasize that the i-th row/column is removed, we assign the corresponding variables as $\widetilde{L}^i$, $\widetilde{\lambda}^i$, etc. The variables corresponding to the original form will be written in standard notation.

The condition number is considered in the form

$$\kappa(\widetilde{L}) = \mathrm{cond}(\widetilde{L}) = \frac{\widetilde{\lambda}_{max}(\widetilde{L})}{\widetilde{\lambda}_{min}(\widetilde{L})}.$$

From one of the variant of Cauchy-like interlacing theorems (see Subsection 2.3.5, Theorem 2.3.25), one can suppose that removing the fixing node does not change the $\widetilde{\lambda}_{max}$ so much as the $\widetilde{\lambda}_{min}$. Therefore, the minimization of the condition number corresponds to the maximization of the $\widetilde{\lambda}_{min}$. Theorem 4.1.1 can be paraphrased into following form:

Theorem 4.1.2 (Maximization of $\widetilde{\lambda}_{min}$)
Removing the vertex k corresponding to the cross-eigenvector center in Laplacian matrix L, the maximal value of $\widetilde{\lambda}^{\star}_{min}$ of $\widetilde{L}$ is obtained over all $\widetilde{\lambda}^{k}_{min}$ of all principal submatrices $\widetilde{L}^k$, i.e., the condition number $\widetilde{L}^k$ is minimized over all k [2].

Remember that the cross-eigenvector center is defined in the node where the corresponding entry of the eigenvector corresponding to the second (and/or third, fourth respectively) smallest eigenvalue of the Laplacian matrix L is minimized in absolute value (i.e., close to zero).

For the sake of simplicity, all eigenvectors are considered in the normalized form, e.g. $\|v_i\| = 1$ ($\|\cdot\|$ is an Euclidean norm).

[2]The symbol $\widetilde{L}^k$ denotes the principal submatrix arising by removing k-th row and column from the original matrix L.

4.2 Preliminaries

From the Rayleigh's principle (Theorem 2.3.11), the smallest eigenvalue of the reduced matrix $\widetilde{L}$ can be computed as

$$0 \neq \widetilde{\lambda}_{min} = \min \frac{x^T \widetilde{L} x}{x^T x}.$$

As we consider the normalized eigenvectors, we can write

$$0 \neq \widetilde{\lambda}_{min} = \min_{\|x\|=1} x^T \widetilde{L} x.$$

The $\widetilde{L}$ is the principal submatrix of matrix L in which some row and column i has been removed. As we would like to find which vertex i from the original matrix L should be removed to obtain the maximal value of $\widetilde{\lambda}_{min}$, we have to rewrite the above equation using original matrix L. Remark that the removing of the vertex i in original matrix L corresponds to setting the corresponding component of vector x to zero, which can be written as

$$0 \neq \widetilde{\lambda}^i_{min} = \min_{\substack{\|x\|=1 \\ x_i=0}} x^T L x.$$

Now, let us substitute the Laplacian matrix by its spectral decomposition. As L is real symmetric matrix (of order $n = |V|$), it has n linearly independent real eigenvectors. Moreover, these eigenvectors can be chosen such that they are orthogonal to each other and have the norm equal to one. The matrix L can be decomposed as

$$L = Q \Lambda Q^T,$$

where Λ is a diagonal matrix of eigenvalues of L ordered in ascending order and Q is the matrix, whose columns are eigenvectors of L ordered accordingly. Thanks to properties of L, $Q^{-1} = Q^T$. Let us further consider e_i to be an n-dimensional unit column vector with one in i-th entry.

$$\begin{aligned} 0 \neq \widetilde{\lambda}^i_{min} &= \min_{\substack{\|x\|=1 \\ x_i=0}} x^T L x \\ &= \min_{\substack{\|x\|=1 \\ x_i=0 \sim x^T e_i=0 \sim e_i^T x=0}} x^T Q \Lambda Q^T x. \end{aligned}$$

Substituting $y = Q^T x$ (therefore $x = Qy$), $y^T = x^T Q$ respectively, we get

$$\begin{aligned} 0 \neq \widetilde{\lambda}^i_{min} &= \min_{\substack{\|y\|=1 \\ e_i^T Q y=0}} y^T \Lambda y \\ &= \min_{\substack{\|y\|=1 \\ r_i(Q) y=0}} y^T \Lambda y. \end{aligned} \tag{4.1}$$

Here, $r_i(Q)$ denotes the i-th row of matrix Q (and it is not the eigenvector). The last equation (Equation (4.2)) could be fully rewritten into y unknown.

$$\text{i.e., } \widetilde{\lambda}^i_{min} = \min_{\substack{\|y\|=1 \\ r_i(Q)y=0}} y^T \Lambda y. \tag{4.2}$$

Finding the maximal value of $\widetilde{\lambda}^\star_{min}$ can be written as

$$\widetilde{\lambda}^\star_{min} = \max_{i=\{1,2,\ldots,n\}} \widetilde{\lambda}^i_{min}. \tag{4.3}$$

4.3 Direct Solution: 1D, $n = 3$

At first, let us try to solve given problem for the simplest example, which is a path with 3 vertices. Let us solve

$$\widetilde{\lambda}^{\star}_{min} = \max_{i=1,2,\dots n} \min_{\substack{\|y\|=1 \\ r_i(Q)y=0}} y^T \Lambda y = \max_{i=1,2,\dots n} \min_{\substack{\|y\|=1 \\ r_i(Q)y=0}} \sum_{i=1}^{n} \lambda_i y_i^2 \ .$$

We can easily set up the Laplacian matrix and its spectral decomposition $L = Q\Lambda Q^T$:

$$L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \ \Lambda = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \text{ and } Q = \begin{bmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} \\ -\frac{\sqrt{3}}{3} & 0 & -\frac{\sqrt{6}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} \end{bmatrix}.$$

As Λ is a diagonal matrix with $\lambda_1, \lambda_2, \lambda_3$, we can substitute into (4.3):

$$\widetilde{\lambda}^{\star}_{min} = \max_{i=1,2,3} \widetilde{\lambda}^{i}_{min} \ , \tag{4.4}$$

$$\widetilde{\lambda}^{i}_{min} = \min_{\substack{y_1^2+y_2^2+y_3^2=1 \\ q_{i1}y_1+q_{i2}y_2+q_{i3}y_3=0}} \mathcal{J}(y_1, y_2, y_3) \ = \tag{4.5}$$

$$= \min_{\substack{y_1^2+y_2^2+y_3^2=1 \\ q_{i1}y_1+q_{i2}y_2+q_{i3}y_3=0}} (y_2^2 + 3y_3^2) \ , \quad \forall i = 1, 2, 3. \tag{4.6}$$

The solution does not depend on y_1 variable, and the conditions form two conditions for two unknowns, so we can express y_2, y_3 from this conditions and substitute into the given problem for each $i = 1, 2, 3$.

In general, we solve this system of equations:

$$\begin{aligned} y_1^2 + y_2^2 + y_3^2 &= 1 \\ q_1y_1 + q_2y_2 + q_3y_3 &= 0 \qquad \text{for} \quad y_2, y_3. \end{aligned} \tag{4.7}$$

We obtain solutions (real, non-zero):

$$y_2 = \frac{-q_1q_2y_1 \pm \sqrt{-q_3^2(q_1^2y_1^2 + q_2^2y_1^2 - q_2^2 + q_3^2y_1^2 - q_3^2)}}{q_2^2 + q_3^2}, \quad q_2^2 + q_3^2 \neq 0,$$

$$y_3 = \frac{-q_1q_3^2y_1 \pm q_2\sqrt{-q_3^2(q_1^2y_1^2 + q_2^2y_1^2 - q_2^2 + q_3^2y_1^2 - q_3^2)}}{q_3(q_2^2 + q_3^2)}, \ q_2^2 + q_3^2 \neq 0, q_3 \neq 0,$$

and we substitute them into functional $\mathcal{J}(y_1, y_2, y_3)$ from (4.5) which means that the functional becomes functional of one variable y_1, i.e., $\mathcal{J}(y_1)$. As the expression of $\mathcal{J}(y_1)$ would be complicated, we write it with certain coefficients q_1, q_2, q_3 in following subsections.

First case $i = 1$:

Solve

$$\begin{aligned} y_1^2 + y_2^2 + y_3^2 &= 1 \\ -\frac{\sqrt{3}}{3}y_1 - \frac{\sqrt{2}}{2}y_2 + \frac{\sqrt{6}}{6}y_3 &= 0 \qquad \text{for} \quad y_2, y_3. \end{aligned}$$

We have obtained two sets of solution, first:

$$\begin{aligned} [y_2]_a &= \frac{1}{4}\left(-\sqrt{6}y_1 - \sqrt{2}\sqrt{2 - 3y_1^2}\right), \\ [y_3]_a &= \frac{1}{4}\left(\sqrt{2}y_1 - \sqrt{6}\sqrt{2 - 3y_1^2}\right), \end{aligned}$$

and second:

$$\begin{aligned} [y_2]_b &= \frac{1}{4}\left(-\sqrt{6}y_1 + \sqrt{2}\sqrt{2 - 3y_1^2}\right), \\ [y_3]_b &= \frac{1}{4}\left(\sqrt{2}y_1 + \sqrt{6}\sqrt{2 - 3y_1^2}\right). \end{aligned}$$

Let us substitute $[y_2]_a, [y_3]_a$ satisfying (4.7) into (4.6)

$$\min_{y_1 \le \sqrt{\frac{2}{3}}} \left(\frac{1}{16}\left(-\sqrt{6}y_1 - \sqrt{2}\sqrt{2 - 3y_1^2}\right)^2 + \frac{3}{16}\left(\sqrt{2}y_1 - \sqrt{6}\sqrt{2 - 3y_1^2}\right)^2 \right).$$

and let us compute gradient y_1:

$$\frac{\partial}{\partial y_1}\left(\frac{1}{16}\left(-\sqrt{6}y_1 - \sqrt{2}\sqrt{2 - 3y_1^2}\right)^2 + \frac{3}{16}\left(\sqrt{2}y_1 - \sqrt{6}\sqrt{2 - 3y_1^2}\right)^2 \right) = 0.$$

We have obtained solution satisfying

$$\frac{3\sqrt{3}y_1^2 - 6y_1\sqrt{2 - 3y_1^2} - \sqrt{3}}{\sqrt{2 - 3y_1^2}} = 0.$$

Stationary points are:

$$\begin{aligned} [y_1]_{a1} &= -\sqrt{\frac{5 - 2\sqrt{5}}{15}} \doteq -0.1876 \ \ldots \text{ local maximizer}, \\ [y_1]_{a2} &= \sqrt{\frac{5 + 2\sqrt{5}}{15}} \doteq 0.7947 \qquad \ldots \text{ local minimizer}. \end{aligned}$$

Using the second set of solution $[y_2]_b, [y_3]_b$ we get:

$$\frac{\partial}{\partial y_1}\left(\frac{1}{16}\left(-\sqrt{6}y_1+\sqrt{2}\sqrt{2-3y_1^2}\right)^2+\frac{3}{16}\left(\sqrt{2}y_1+\sqrt{6}\sqrt{2-3y_1^2}\right)^2\right)=0.$$

We have obtained solution satisfying

$$\frac{-3\sqrt{3}y_1^2-6y_1\sqrt{2-3y_1^2}+\sqrt{3}}{\sqrt{2-3y_1^2}}=0.$$

Stationary points are:

$$\begin{aligned}[y_1]_{b1} &= -\sqrt{\frac{5+2\sqrt{5}}{15}}\doteq -0.7947 \ \ldots \text{ local minimizer},\\ [y_1]_{b2} &= \sqrt{\frac{5-2\sqrt{5}}{15}}\doteq 0.1876 \qquad \ldots \text{ local maximizer}.\end{aligned}$$

We get different solution of (4.6) for every set $[y_1]_{a2}$, $[y_2]_a$, $[y_3]_a$, and $[y_1]_{b1}$, $[y_2]_b$, $[y_3]_b$:

$$\begin{aligned}[\widetilde{\lambda}^1_{min}]_{a2} &= \frac{3-\sqrt{5}}{2}\doteq 0.3820,\\ [\widetilde{\lambda}^1_{min}]_{b1} &= \frac{3(5-\sqrt{5})}{10}\doteq 0.8292.\end{aligned}$$

The minimal value $[\widetilde{\lambda}^1_{min}]_{a2}=\frac{3-\sqrt{5}}{2}\doteq 0.3820$ **occurs in** $[y_1]_{a2}=\sqrt{\frac{5+2\sqrt{5}}{15}}\doteq 0.7947$**.**

Second case $i=2$**:**

We get another system of equations with choice $i=2$:

$$\begin{aligned}y_1^2+y_2^2+y_3^2 &= 1\\ -\frac{\sqrt{3}}{3}y_1+0\cdot y_2+\frac{\sqrt{6}}{3}y_3 &= 0 \qquad \text{for} \quad y_2, y_3.\end{aligned}$$

We have obtained solutions:

$$\begin{aligned}y_2 &= \pm\sqrt{1-\frac{3y_1^2}{2}},\\ y_3 &= \frac{y_1}{\sqrt{2}}.\end{aligned}$$

Let us substitute y_2, y_3 into (4.6)

$$\min_{y_1\in\mathbb{R}}\left(\left(\pm\sqrt{1-\frac{3y_1^2}{2}}\right)^2+3\left(\frac{y_1}{\sqrt{2}}\right)^2\right)=\min_{y_1\in\mathbb{R}}1=1=\lambda_2. \qquad (4.8)$$

In this case, we get the minimal value $\widetilde{\lambda}^2_{min}=1$.

Third case $i=3$:

Using $i=3$, we get system of equations similar to first case:

$$\begin{aligned} y_1^2+y_2^2+y_3^2 &= 1\\ -\frac{\sqrt{3}}{3}y_1+\frac{\sqrt{2}}{2}y_2+\frac{\sqrt{6}}{6}y_3 &= 0 \qquad \text{for} \quad y_2, y_3. \end{aligned}$$

Again, we have obtained two sets of solution, first:

$$\begin{aligned} [y_2]_a &= \frac{1}{4}\left(\sqrt{6}y_1-\sqrt{2}\sqrt{2-3y_1^2}\right),\\ [y_3]_a &= \frac{1}{4}\left(\sqrt{2}y_1+\sqrt{6}\sqrt{2-3y_1^2}\right), \end{aligned}$$

and second:

$$\begin{aligned} [y_2]_b &= \frac{1}{4}\left(\sqrt{6}y_1+\sqrt{2}\sqrt{2-3y_1^2}\right),\\ [y_3]_b &= \frac{1}{4}\left(\sqrt{2}y_1-\sqrt{6}\sqrt{2-3y_1^2}\right). \end{aligned}$$

After substitution $[y_2]_a, [y_3]_a$ and $[y_2]_b, [y_3]_b$ into (4.6) and after computing gradient, we obtain four different stationary points $[y_1]_{a1}, [y_1]_{a2}, [y_1]_{b1}, [y_1]_{b2}$ (the same as in first case):

$$\begin{aligned} [y_1]_{a1} &= -\sqrt{\frac{5+2\sqrt{5}}{15}} \doteq -0.7947 \ \ldots \text{ local minimizer},\\ [y_1]_{a2} &= \sqrt{\frac{5-2\sqrt{5}}{15}} \doteq 0.1876 \qquad \ldots \text{ local maximizer},\\ [y_1]_{b1} &= -\sqrt{\frac{5-2\sqrt{5}}{15}} \doteq -0.1876 \ \ldots \text{ local minimizer},\\ [y_1]_{b2} &= \sqrt{\frac{5+2\sqrt{5}}{15}} \doteq 0.7947 \qquad \ldots \text{ local maximizer}. \end{aligned}$$

Again, we obtain solutions of (4.6) for every set $[y_1]_{a1}$, $[y_2]_a$, $[y_3]_a$, and $[y_1]_{b1}$, $[y_2]_b$, $[y_3]_b$:

$$\begin{aligned}[\widetilde{\lambda}^3_{min}]_{a1} &= \frac{3(5-\sqrt{5})}{10} \doteq 0.8292,\\ [\widetilde{\lambda}^3_{min}]_{b2} &= \frac{3-\sqrt{5}}{2} \doteq 0.3820.\end{aligned}$$

The minimal value $[\widetilde{\lambda}^3_{min}]_{b2} = \frac{3-\sqrt{5}}{2} \doteq 0.3820$ **occurs in** $[y_1]_{b2} = \sqrt{\frac{5+2\sqrt{5}}{15}} \doteq 0.7947$**.**

Conclusion

Finally, we find the maximum over all $i = 1, 2, 3$ (4.4):

$$\widetilde{\lambda}^\star_{min} = \max_{i=1,2,3} \widetilde{\lambda}^i_{min} = \max\left\{\frac{3-\sqrt{5}}{2}, 1, \frac{3-\sqrt{5}}{2}\right\}.$$

Corollary 4.3.1 *The maximal solution over all i occurs in $i = 2$, $\widetilde{\lambda}^\star_{min} = \widetilde{\lambda}^2_{min} = 1$.*

4.3.1 Geometrical view

Let us remind equations for $n = 3$ and for given Λ:

$$\widetilde{\lambda}^i_{min} = \min_{\substack{y_1^2+y_2^2+y_3^2=1\\ q_{i1}y_1+q_{i2}y_2+q_{i3}y_3=0}} (y_2^2 + 3y_3^2)\ , \quad \forall i = 1, 2, 3. \tag{4.9}$$

Thanks to Cauchy Interlacing Theorem 2.3.25,

$$0 = \lambda_{1=min} \le \widetilde{\lambda}_{1=min} \le \lambda_2 \le \ldots \le \widetilde{\lambda}_{n-1} \le \lambda_n,$$

we get the upper bound for $\widetilde{\lambda}^i_{min} \le \lambda_2$. In Section 4.3, we have shown that for given $q_{i2} = 0$ we obtain exactly the upper bound:

$$\widetilde{\lambda}^i_{min} = \min_{\substack{y_1^2+y_2^2+y_3^2=1\\ q_{i1}y_1+q_{i2}y_2+q_{i3}y_3=0\\ q_{i2}=0}} (y_2^2 + 3y_3^2) = \lambda_2.$$

Let us have a look on problem (4.9) in three-dimensional view. The condition $y_1^2 + y_2^2 + y_3^2 = 1$ forms a sphere, meanwhile the condition $q_{i1}y_1 + q_{i2}y_2 + q_{i3}y_3 = 0$ is a plane. We are finding such a vector q_i for which

the minimum of (4.9) will be maximal (over all $i = 1, 2, 3$). Remark that $q_{i1} = \pm\frac{1}{\sqrt{3}}$ (first eigenvector is a normalized constant vector). Let us show that $q_{i2} = 0$.

In Figures 4.1-4.3, there are depicted values of function $\mathcal{J}(y_1, y_2, y_3) = y_2^2 + 3y_3^2$ on sphere $y_1^2 + y_2^2 + y_3^2 = 1$ together with plane $q_{i1}y_1 + q_{i2}y_2 + q_{i3}y_3 = 0$ for $i = 1, 2, 3$. Especially, in Figure 4.1, there is plotted the plane $-\frac{\sqrt{3}}{3}y_1 + 0 \cdot y_2 + \frac{\sqrt{6}}{3}y_3 = 0$ (it does not depend on y_2).

Considering $i = 2$ (such as $q_{i2} = 0$), let us notice several interesting observations:

- Function $\mathcal{J}(y_1, y_2, y_3) = y_2^2 + 3y_3^2$ does not depend on y_1. It acquires the minimum $\mathcal{J}(y_1, y_2, y_3) = 0 = \lambda_1$ in point $[0, 0, 0]$ and along y_1 axis, i.e., in points $[\pm 1, 0, 0]$ on unit sphere.
- In points $[\pm 1, 0, 0]$, $[0, \pm 1, 0]$ and $[0, 0, \pm 1]$ it is equal to λ_1, λ_2, and λ_3 respectively.
- The intersection of the sphere and plane $q_{i1}y_1 + q_{i2}y_2 + q_{i3}y_3 = 0$ is a circle with a center in point $[0, 0, 0]$ for each choice of q_{i1}, q_{i2}, q_{i3}.
- Points $[\pm 1, 0, 0]$ (where the function attains the global minimum) belong to circle corresponding to intersection of the sphere with the plane y_1y_2 (i.e. $y_3 = 0$). In this circle, the function attains its maximum $\lambda_2 = 1$ in $[0, \pm 1, 0]$.
- Every plane $q_{i1}y_1 + q_{i2}y_2 + q_{i3}y_3 = 0$, in which we find the minimum of $\mathcal{J}$ has to cross this circle. Thus, the minimum is at most $\lambda_2 = 1$.
- For given $q_{i2} = 0$, we show that the minimum is right $\lambda_2 = 1$: If $q_{i2} = 0 \Rightarrow cos\beta = 0$, where β is an angle between normal line of plane $q_{i1}y_1 + q_{i2}y_2 + q_{i3}y_3 = 0$ and y_2 axis. y_2 axis is perpendicular to normal line of the plane, i.e., y_2 axis lies in the plane. The normal line of the plane is the row r_i, i.e., $\overrightarrow{n} = (q_{i1}, q_{i2}, q_{i3}) = \left(-\frac{\sqrt{3}}{3}, 0, \frac{\sqrt{6}}{3}\right)$ for $i = 2$. We obtain two direction vectors, e.g., $\overrightarrow{u} = \left[\frac{\sqrt{6}}{3}, 0, \frac{\sqrt{3}}{3}\right]$ and $\overrightarrow{v} = [0, 1, 0]$.
- In this case, function $\mathcal{J}(y_1, y_2, y_3)$ is constantly equal to $\lambda_2 = 1$ along the circle obtained as intersection of the sphere and plane $-\frac{\sqrt{3}}{3}y_1 + 0 \cdot y_2 + \frac{\sqrt{6}}{3}y_3 = 0$ (with $q_{i2} = 0$).
- Therefore, the minimum along this circle is also equal to $\lambda_2 = 1$.
- The maximum over all $i = 1, 2, 3$ is equal to $\lambda_2 = 1$ and it cannot be extended due to Cauchy interlacing theorem 2.3.25.

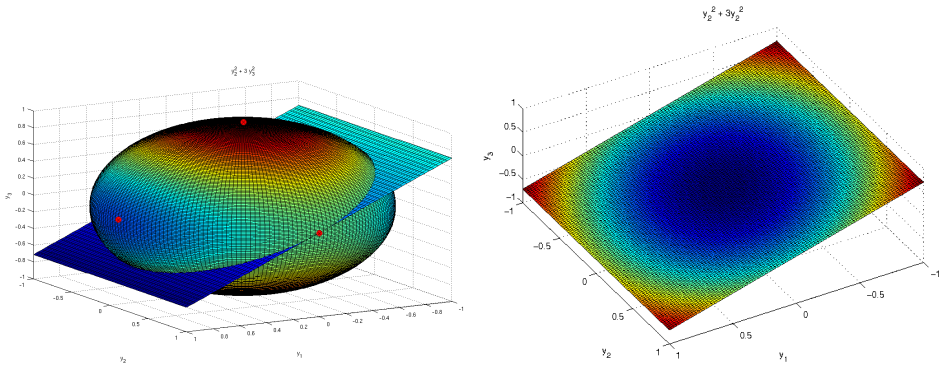


Figure 4.1: Intersection of a sphere and a plane $(-\frac{\sqrt{3}}{3}y_1 + 0 \cdot y_2 + \frac{\sqrt{6}}{3}y_3 = 0)$

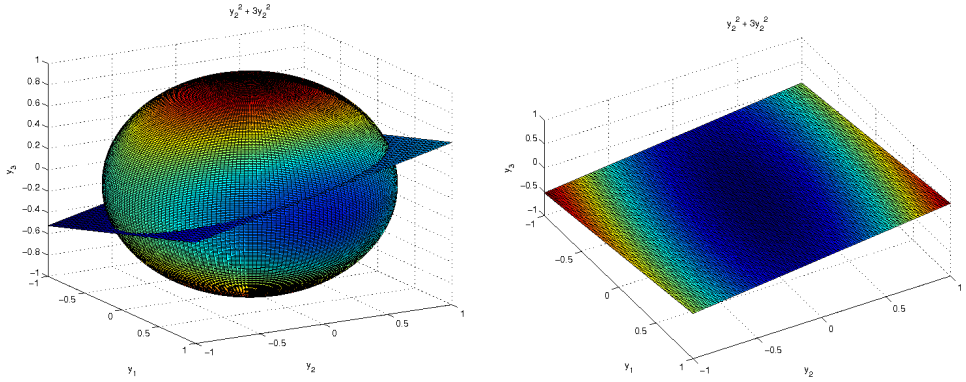


Figure 4.2: Intersection of a sphere and a plane $(-\frac{\sqrt{3}}{3}y_1 - \frac{\sqrt{2}}{2}y_2 + \frac{\sqrt{6}}{6}y_3 = 0)$

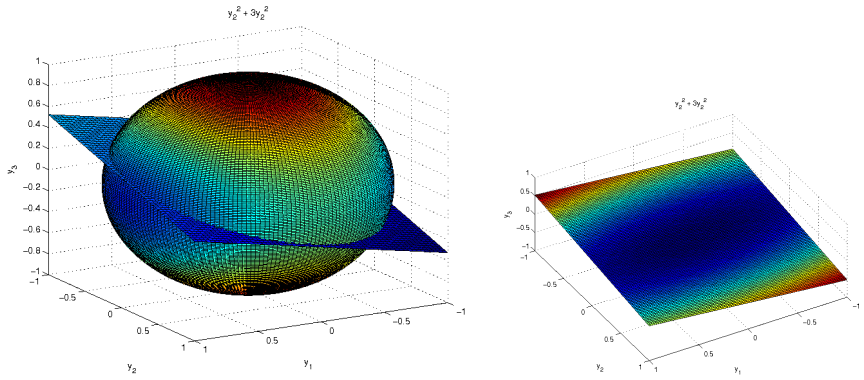


Figure 4.3: Intersection of a sphere and a plane $(-\frac{\sqrt{3}}{3}y_1 + \frac{\sqrt{2}}{2}y_2 + \frac{\sqrt{6}}{6}y_3 = 0)$

4.4 Solution: 1D, extension to n variables

Consider an one–dimensional example, especially P_n in graph theory meaning. In following subsections, we will use several theorems from Section 2.3.

4.4.1 Preliminaries

We solve following system of equation (for given n):

$$\begin{aligned}
\widetilde{\lambda}^{\star}_{min} &= \max_{i=1,2,\ldots,n} \widetilde{\lambda}^{i}_{min}, \\
\widetilde{\lambda}^{i}_{min} &= \min \mathcal{J}(y_1, y_2, \ldots, y_n) \\
&= \min \left(\lambda_1 y_1^2 + \lambda_2 y_2^2 + \quad \cdots \quad + \lambda_n y_n^2 \right) \qquad (4.10) \\
&\quad \text{subject to} \\
&\quad y_1^2 + y_2^2 + \cdots + y_n^2 \qquad = 1 \qquad (4.11) \\
&\quad q_{i1} y_1 + q_{i2} y_2 + \cdots + q_{in} y_n \qquad = 0. \qquad (4.12)
\end{aligned}$$

Again, thanks to Cauchy Interlacing Theorem 2.3.25, $0 = \lambda_{1=min} \leq \widetilde{\lambda}_{1=min} \leq \lambda_2 \leq \ldots \leq \widetilde{\lambda}_{n-1} \leq \lambda_n$, we get the upper bound for $\widetilde{\lambda}^{i}_{min} \leq \lambda_2$. We will show, that for i such that $q_{i2} = 0$ we obtain exactly the upper bound:

$$\widetilde{\lambda}^{i}_{min} \;=\; \min \left(\lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2 \right) = \lambda_2.$$

4.4.2 Eigenvalues and Eigenvectors

Let us look on the exact form of eigenvalues and eigenvectors of P_n from Section 2.4:

$$\lambda_k \;=\; 2 - 2\cos\left(\frac{\pi(k-1)}{n}\right), \qquad k = 1, 2, \ldots, n, \qquad (4.13)$$

$$v_k[i] \;=\; \cos\left(\frac{\pi(k-1)}{n}\left(i - \frac{1}{2}\right)\right), \quad k, i = 1, 2, \ldots, n. \qquad (4.14)$$

Unfortunately, equality (4.14) does not provide the eigenvectors in normed form. We have to compute norm for each v_k. For each k we get:

$$k = 1: \quad \|v_1\| \;=\; \sqrt{\sum_{i=1}^{n} \cos^2(0)} = \sqrt{n}, \qquad (4.15)$$

$$\forall k \in \langle 2, n \rangle : \|v_k\| \;=\; \sqrt{\sum_{i=1}^{n} \cos^2\left(\frac{\pi(k-1)}{n}\left(i - \frac{1}{2}\right)\right)} = \sqrt{\frac{n}{2}}. \qquad (4.16)$$

The last sum (4.16) was computed using Maple software ([48]). The normalized eigenvectors then take the form

$$\begin{aligned} v_k[i] &= \frac{\cos\left(\frac{\pi(k-1)}{n}\left(i-\frac{1}{2}\right)\right)}{\sqrt{\sum_{i=1}^{n}\cos^2\left(\frac{\pi(k-1)}{n}\left(i-\frac{1}{2}\right)\right)}}, \quad k=1,2,\ldots,n, \\ v_1[i] &= \frac{1}{\sqrt{n}}, \qquad (4.17) \\ v_k[i] &= \frac{\sqrt{2}}{\sqrt{n}}\cos\left(\frac{\pi(k-1)}{n}\left(i-\frac{1}{2}\right)\right), \quad k=2,\ldots,n. \qquad (4.18) \end{aligned}$$

Here, $v_k[i]$ corresponds to q_{ik}, i.e., if we consider the row of matrix Q in which the $q_{i2}=0$, we find such i for which the $v_2[i]=0$. Remark that if the v_k is a normalized eigenvector then the $-v_k$ is also a normalized eigenvector.

$$\begin{aligned} v_2[i]=\cos\left(\frac{\pi}{n}\left(i-\frac{1}{2}\right)\right) &= 0 \\ \frac{\pi}{n}\left(i-\frac{1}{2}\right) &= \frac{\pi}{2}(2p+1), \quad p\in\mathbb{Z} \\ i-\frac{1}{2} &= \left(p+\frac{1}{2}\right)n, \quad i\in\langle 1,n\rangle \Rightarrow p=0 \\ i &= \frac{n+1}{2} \qquad \text{(n is odd).} \qquad (4.19) \end{aligned}$$

Remark that n has to be odd to obtain integer i and, moreover, $q_{i2}=0$ for integer i corresponding to odd n only!

Using informations (4.17), (4.18), and (4.19), we can set up the exact form of (4.12) for $i=\frac{n+1}{2}$:

$$\begin{aligned} \frac{1}{\sqrt{n}}y_1+\sum_{k=2}^{n}\left(\frac{\sqrt{2}}{\sqrt{n}}\cos\left(\frac{\pi(k-1)}{2}\right)y_k\right) &= 0 \\ \frac{1}{\sqrt{n}}y_1+\frac{\sqrt{2}}{\sqrt{n}}\cos\left(\frac{\pi}{2}\right)y_2+\frac{\sqrt{2}}{\sqrt{n}}\cos(\pi)\,y_3+ & \\ +\frac{\sqrt{2}}{\sqrt{n}}\cos\left(\frac{3\pi}{2}\right)y_4+\cdots+\frac{\sqrt{2}}{\sqrt{n}}\cos\left(\frac{(n-1)\pi}{2}\right)y_n &= 0, \\ \frac{1}{\sqrt{n}}y_1-\frac{\sqrt{2}}{\sqrt{n}}y_3+\frac{\sqrt{2}}{\sqrt{n}}y_5+\cdots+(-1)^{\frac{(n-1)}{2}}\frac{\sqrt{2}}{\sqrt{n}}y_n &= 0. \qquad (4.20) \end{aligned}$$

Last equation (4.20) is a general equation of plane ($\overrightarrow{n} \cdot \overrightarrow{y} = 0$) with the normal vector $\overrightarrow{n}$

$$\overrightarrow{n} = \left[\frac{1}{\sqrt{n}}, 0, -\frac{\sqrt{2}}{\sqrt{n}}, 0, \frac{\sqrt{2}}{\sqrt{n}}, 0, -\frac{\sqrt{2}}{\sqrt{n}}, \ldots, 0, (-1)^{\frac{(n-1)}{2}} \frac{\sqrt{2}}{\sqrt{n}}\right]^T.$$

Let us make a substitution $s = \frac{\sqrt{2}}{\sqrt{n}}$. Using this substitution we obtain

$$\overrightarrow{n} = \left[\frac{s}{\sqrt{2}}, 0, -s, 0, s, 0, -s, \ldots, 0, (-1)^{\frac{(n-1)}{2}} s\right]^T.$$

4.4.3 Parametric Equation of the Plane

The general equation of plane is not suitable for finding $y_1, y_3, \ldots, y_n$. However, we can easily set up $(n-1)$ n-dimensional direction vectors perpendicular to normal vector $\overrightarrow{n}$ which will be more suitable for our purposes:

$$\overrightarrow{p_2} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \overrightarrow{p_4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \ldots, \quad \overrightarrow{p}_{2k} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}, k = \frac{n-1}{2},$$

$$\overrightarrow{p_1} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} \\ 0 \\ \frac{1}{\sqrt{3}} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \overrightarrow{p_3} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{3}} \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \ldots, \overrightarrow{p}_{2k-1} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \frac{1}{\sqrt{3}} \end{bmatrix}, k = \frac{n-1}{2}.$$

Let us set up a parametric equation of the plane (4.20) in new variables $a_1, a_2, \ldots, a_{n-1}$:

$$P(a_1, a_2, \ldots, a_{n-1}) = \quad a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots + a_{n-1}\overrightarrow{p}_{n-1} = \overrightarrow{y}.$$

$$P(a_1, a_2, \ldots, a_{n-1}) = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}}(a_1 + a_3 + \cdots + a_{n-1}) \\ a_2 \\ \frac{1}{\sqrt{3}}a_1 \\ a_4 \\ \frac{1}{\sqrt{3}}a_3 \\ \vdots \\ \frac{1}{\sqrt{3}}a_{n-4} \\ a_{n-1} \\ \frac{1}{\sqrt{3}}a_{n-2} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}.$$

Now, let us substitute solution $y_1, y_2, \ldots, y_n$ into functional $\mathcal{J}(y_1, y_2, \ldots, y_n)$ from (4.10) and into condition (4.11). We obtain the functional in new $(n-1)$ variables $a_1, a_2, \ldots, a_{n-1}$. Consider $\lambda_1 = 0$, we obtain

$$\mathcal{J}(a_1, a_2, \ldots, a_{n-1}) = \lambda_2 a_2^2 + \frac{\lambda_3}{3}a_1^2 + \lambda_4 a_4^2 + \cdots + \frac{\lambda_n}{3}a_{n-2}^2 \tag{4.21}$$

subject to

$$\frac{2}{3}(a_1 + a_3 + \cdots + a_{n-1})^2 + a_2^2 + \frac{1}{3}a_1^2 + a_4^2 + \cdots + a_{n-1}^2 + \frac{1}{3}a_{n-2}^2 = 1. \tag{4.22}$$

4.4.4 Lagrange Multiplier

To solve system (4.21) with single constrain (4.22), let us use Lagrange multiplier for finding minimum of $\mathcal{J}(a_1, a_2, \ldots, a_{n-1})$ subject to condition (4.22):

$$\Lambda(a_1, a_2, \ldots, a_{n-1}, \mu) = \mathcal{J}(a_1, a_2, \ldots, a_{n-1}) - \mu(g(a_1, a_2, \ldots, a_{n-1}) - 1),$$

where

$$\begin{aligned} \mathcal{J}(a_1, a_2, \ldots, a_{n-1}) &= \lambda_2 a_2^2 + \frac{\lambda_3}{3}a_1^2 + \lambda_4 a_4^2 + \cdots + \frac{\lambda_n}{3}a_{n-2}^2 \quad \text{and} \\ g(a_1, a_2, \ldots, a_{n-1}) &= \frac{2}{3}(a_1 + a_3 + \cdots + a_{n-1})^2 + a_2^2 + \frac{1}{3}a_1^2 + a_4^2 + \cdots + \\ &\quad + a_{n-1}^2 + \frac{1}{3}a_{n-2}^2. \end{aligned}$$

We get following system of equation:

$$\begin{cases} g(a_1, a_2, \ldots, a_{n-1}) = 1, \\ \nabla\mathcal{J}(a_1, a_2, \ldots, a_{n-1}) - \mu\nabla g(a_1, a_2, \ldots, a_{n-1}) = 0. \end{cases}$$

In our particular example:

$$\begin{aligned}
g(a_1,a_2,\ldots,a_{n-1}) = \quad & \frac{2}{3}(a_1+a_3+\cdots+a_{n-1})^2+a_2^2+\frac{1}{3}a_1^2+a_4^2+\cdots+ \\
& +a_{n-1}^2+\frac{1}{3}a_{n-2}^2 = \quad 1, \\
\frac{\partial\mathcal{J}}{\partial a_1}-\mu\frac{\partial g}{\partial a_1} = \quad & \frac{2}{3}\lambda_3 a_1-\mu\left(\frac{2}{3}a_1+\frac{4}{3}(a_1+a_3+\cdots+a_{n-2})\right) = \quad 0, \\
\frac{\partial\mathcal{J}}{\partial a_2}-\mu\frac{\partial g}{\partial a_2} = \quad & 2\lambda_2 a_2-\mu 2a_2 = \quad 0, \\
\frac{\partial\mathcal{J}}{\partial a_3}-\mu\frac{\partial g}{\partial a_3} = \quad & \frac{2}{3}\lambda_5 a_3-\mu\left(\frac{2}{3}a_3+\frac{4}{3}(a_1+a_3+\cdots+a_{n-2})\right) = \quad 0, \\
\frac{\partial\mathcal{J}}{\partial a_4}-\mu\frac{\partial g}{\partial a_4} = \quad & 2\lambda_4 a_4-\mu 2a_4 = \quad 0, \\
\vdots \qquad & \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
\frac{\partial\mathcal{J}}{\partial a_{n-2}}-\mu\frac{\partial g}{\partial a_{n-2}} = \quad & \frac{2}{3}\lambda_n a_{n-2}-\mu\left(\frac{2}{3}a_{n-2}+\frac{4}{3}(a_1+a_3+\cdots+a_{n-2})\right) = \quad 0, \\
\frac{\partial\mathcal{J}}{\partial a_{n-1}}-\mu\frac{\partial g}{\partial a_{n-1}} = \quad & 2\lambda_{n-1}a_{n-1}-\mu 2a_{n-1} = \quad 0.
\end{aligned}$$

Let us solve this system of equations for n variables $a_1, a_2, \ldots, a_{n-1}$, and μ. As μ is Lagrange multiplier, we suppose $\mu \neq 0$. Let us discuss all possible variant of solution:

1. $\mu = \lambda_{2k}$ $(k = \{1, 2, \ldots, \frac{n-1}{2}\})$:
   $a_{2k} \in \langle -1, 1\rangle, a_{2l} = 0, \forall l \neq k$ (to satisfy all condition for even a together with condition for $g(a_1, a_2, \ldots, a_{n-1})$),
   (a) $a_{2k} = \pm 1$,
   (b) $a_{2k} \in (-1, 1)$,
2. $\mu \neq \lambda_{2k}$ $(k = \{1, 2, \ldots, \frac{n-1}{2}\})$:
   $a_{2k} = 0, \forall k$ (to satisfy all condition for even a).

ad 1.a) Variant $\mu = \lambda_{2k}$, $a_{2k} = \pm 1$:

If $a_{2k} = \pm 1$, all remaining variables has to be zero, i.e., $a_l = 0\ \forall l \neq 2k$, $k = \{1, 2, \ldots, \frac{n-1}{2}\}$, to satisfy the first condition ($g(a_1, a_2, \ldots, a_{n-1}) = 1$).

After substitution all variables into (4.21) we get

$$\mathcal{J}(a_1, a_2, \ldots, a_{n-1}) = \lambda_{2k}.$$

As we know, that the new $\widetilde{\lambda}^i_{min} \leq \lambda_2$, we get $k = 1$ and

$$\mathcal{J}(a_1, a_2, \ldots, a_{n-1}) = \lambda_2.$$

□

ad 1.b) Variant $\mu = \lambda_{2k}$, $a_{2k} \in (-1, 1)$:

Variables a_{2k-1} assigned as a_t can be computed by subtraction of condition (1) from condition (t):

$$a_{2k-1} = a_t = \frac{a_1(\lambda_3 - \mu)}{\lambda_{t+1} - \mu} = \frac{a_1(\lambda_3 - \lambda_{2k})}{\lambda_{t+2} - \lambda_{2k}}.$$

We get all odd variables depending on a_1, all even variables are equal to zero except a_{2k} and $\mu = \lambda_{2k}$. To compute missing variable a_{2k} we substitute all computed variables into condition (norm):

$$a_{2k}^2 = 1 - \frac{2}{3}a_1^2(\lambda_3 - \lambda_{2k})^2 \left[\frac{1}{\lambda_3 - \lambda_{2k}} + \frac{1}{\lambda_5 - \lambda_{2k}} + \cdots + \frac{1}{\lambda_n - \lambda_{2k}}\right]^2 - \frac{1}{3}a_1^2(\lambda_3 - \lambda_{2k})^2 \left[\frac{1}{(\lambda_3 - \lambda_{2k})^2} + \frac{1}{(\lambda_5 - \lambda_{2k})^2} + \cdots + \frac{1}{(\lambda_n - \lambda_{2k})^2}\right].$$

Now, let us substitute all computed variables into $\mathcal{J}(a_1, a_2, \ldots, a_{n-1})$:

$$\begin{aligned}\mathcal{J}(a_1, a_2, \ldots, a_{n-1}) &= \lambda_2 a_2^2 + \frac{1}{3}\lambda_3 a_1^2 + \lambda_4 a_4^2 + \cdots + \frac{1}{3}\lambda_n a_{n-2}^2 \\ &= \lambda_{2k} a_{2k}^2 + \frac{1}{3}\lambda_3 a_1^2 + \frac{1}{3}\lambda_5 a_3^2 + \cdots + \frac{1}{3}\lambda_n a_{n-2}^2 \ ,\end{aligned}$$

$$\begin{aligned}\mathcal{J}(a_1, a_{2k}) &= \lambda_{2k} a_{2k}^2 + \frac{1}{3}\lambda_3 a_1^2 + \frac{1}{3}\lambda_5 a_1^2 \frac{(\lambda_3 - \lambda_{2k})^2}{(\lambda_5 - \lambda_{2k})^2} + \frac{1}{3}\lambda_7 a_1^2 \frac{(\lambda_3 - \lambda_{2k})^2}{(\lambda_7 - \lambda_{2k})^2} + \\ &\quad + \cdots + \frac{1}{3}\lambda_n a_1^2 \frac{(\lambda_3 - \lambda_{2k})^2}{(\lambda_n - \lambda_{2k})^2} \ , \\ \mathcal{J}(a_1, a_{2k}) &= \lambda_{2k} a_{2k}^2 + \frac{1}{3}a_1^2(\lambda_3 - \lambda_{2k})^2 \left[\frac{\lambda_3}{(\lambda_3 - \lambda_{2k})^2} + \frac{\lambda_5}{(\lambda_5 - \lambda_{2k})^2} + \right. \\ &\quad \left. + \frac{\lambda_7}{(\lambda_7 - \lambda_{2k})^2} + \cdots + \frac{\lambda_n}{(\lambda_n - \lambda_{2k})^2}\right] \ ,\end{aligned}$$

$$\mathcal{J}(a_1) = \lambda_{2k} + \frac{1}{3}a_1^2(\lambda_3 - \lambda_{2k})^2\Big[\bigstar\Big] \ ,$$

$$\begin{aligned}
\bigstar = & \left(\frac{\lambda_3}{(\lambda_3-\lambda_{2k})^2} + \frac{\lambda_5}{(\lambda_5-\lambda_{2k})^2} + \cdots + \frac{\lambda_n}{(\lambda_n-\lambda_{2k})^2}\right) - \\
& - \lambda_{2k}\left(\frac{1}{(\lambda_3-\lambda_{2k})^2} + \frac{1}{(\lambda_5-\lambda_{2k})^2} + \cdots + \frac{1}{(\lambda_n-\lambda_{2k})^2}\right) - \\
& - 2\lambda_{2k}\left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right)^2 \\
\bigstar = & \left(\frac{\lambda_3-\lambda_{2k}}{(\lambda_3-\lambda_{2k})^2} + \frac{\lambda_5-\lambda_{2k}}{(\lambda_5-\lambda_{2k})^2} + \cdots + \frac{\lambda_n-\lambda_{2k}}{(\lambda_n-\lambda_{2k})^2}\right) - \\
& - 2\lambda_{2k}\left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right)^2 , \\
\bigstar = & \left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right) - \\
& - 2\lambda_{2k}\left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right)^2 , \\
\bigstar = & \left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right) \cdot \\
& \left(1 - 2\lambda_{2k}\left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right)\right) .
\end{aligned}$$

Using Maple software ([48]) together with information $\lambda_k = 2 - 2\cos\left(\frac{\pi(k-1)}{n}\right), k = 1, 2, \ldots, n$ we get

$$2\lambda_{2k}\left(\frac{1}{\lambda_3-\lambda_{2k}} + \frac{1}{\lambda_5-\lambda_{2k}} + \cdots + \frac{1}{\lambda_n-\lambda_{2k}}\right) = 1 \ .$$

Therefore,

$$\begin{aligned}
\mathcal{J}(a_1) &= \lambda_{2k} + \frac{1}{3}a_1^2(\lambda_3 - \lambda_{2k})^2\big[0\big] \ , \\
\mathcal{J}(a_1) &= \lambda_{2k} \ .
\end{aligned}$$

As we know, that the new $\widetilde{\lambda}^i_{min} \leq \lambda_2$, we get $k = 1$ and

$$\mathcal{J}(a_1) = \lambda_2 \ .$$

□

ad 2.) Variant $\mu \neq \lambda_{2k}$, $a_{2k} = 0$:

Again, variables a_{2k-1} assigned as a_t can be computed by subtraction of condition (1) from condition (t):

$$a_{2k-1} = a_t = \frac{a_1(\lambda_3 - \mu)}{\lambda_{t+2} - \mu}.$$

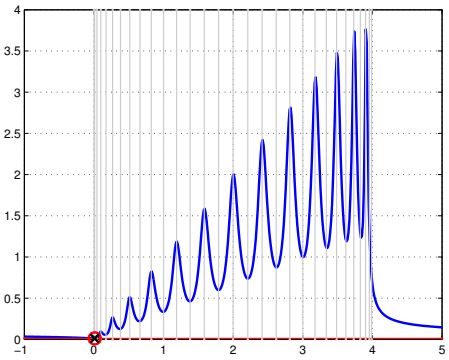
We get all odd variables depending on a_1, all even variables are equal to zero. To compute missing variable μ, we substitute all computed variables into the first condition for the norm:

$$f(a_1, \mu) = 1 - \frac{2}{3}a_1^2(\lambda_3 - \mu)^2 \left[\frac{1}{\lambda_3 - \mu} + \frac{1}{\lambda_5 - \mu} + \cdots + \frac{1}{\lambda_n - \mu}\right]^2 - \frac{1}{3}a_1^2(\lambda_3 - \mu)^2 \left[\frac{1}{(\lambda_3 - \mu)^2} + \frac{1}{(\lambda_5 - \mu)^2} + \cdots + \frac{1}{(\lambda_n - \mu)^2}\right] = 0.$$

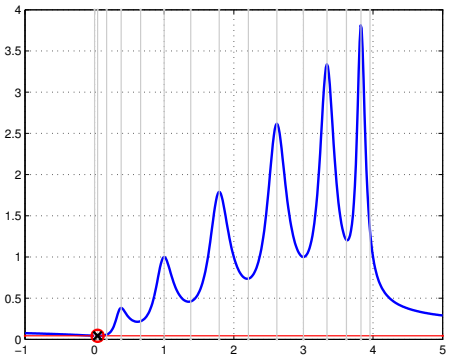
Function $f(a_1, \mu)$ is an implicit function for μ. Extraction of μ is very complicated. However, we can compute μ numerically and substitute into (4.21). We obtain function of one variable $\mathcal{J}(a_1, \mu)$ (for given μ).

In Figures 4.4, there is depicted function $\mathcal{J}(a_1, \mu)$ with blue line for different choice of n. Grey vertical lines correspond to $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. We can see that the minimum of $\mathcal{J}(a_1, \mu)$ occurs in the point corresponding to λ_2. The function value of the minimum is equal to λ_2 (depicted with red line).

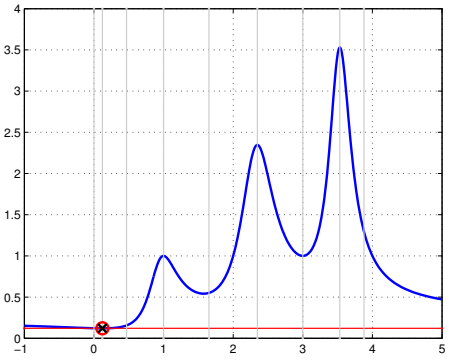
□



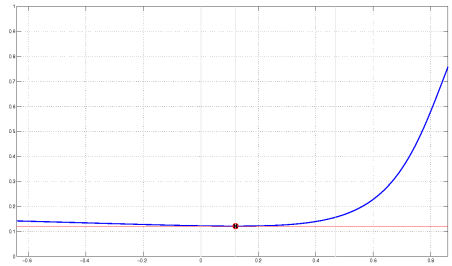
(a) $n = 30$



(b) $n = 15$



(c) $n = 9$



(d) $n = 9$ (zoom)

Figure 4.4: Graph $\mathcal{J}(a_1, \mu)$

4.5 Solution: extension into 2D

Let us extend our consideration into two–dimensions. Let us consider a two–dimensional example, especially Cartesian product $P_n \times P_m$ in graph theory meaning. We solve the same system of equations (for given n) as in one–dimension:

$$\begin{aligned}
\widetilde{\lambda}^{\star}_{min} &= \max_{i=1,2,\ldots,n} \widetilde{\lambda}^{i}_{min}, \\
\widetilde{\lambda}^{i}_{min} &= \min \mathcal{J}(y_1, y_2, \ldots, y_n) \\
&= \min \left(\lambda_1 y_1^2 + \lambda_2 y_2^2 + \quad \cdots \quad + \lambda_n y_n^2 \right) \qquad (4.23) \\
&\quad \text{subject to} \\
&\quad y_1^2 + y_2^2 + \cdots + y_n^2 \qquad = 1 \qquad (4.24) \\
&\quad q_{i1} y_1 + q_{i2} y_2 + \cdots + q_{in} y_n \qquad = 0. \qquad (4.25)
\end{aligned}$$

Thanks to Cauchy Interlacing Theorem 2.3.25, $0 = \lambda_{1=min} \leq \widetilde{\lambda}_{1=min} \leq \lambda_2 \leq \ldots \leq \widetilde{\lambda}_{n-1} \leq \lambda_n$, we again get the upper bound for $\widetilde{\lambda}^{i}_{min} \leq \lambda_2$. We will show that for i such that $q_{i2} = 0$ we obtain exactly the upper bound:

$$\widetilde{\lambda}^{i}_{min} \;=\; \min \left(\lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2 \right) = \lambda_2.$$

In two–dimensions, we use properties of Cartesian product $P_n \times P_m$. As we are able to compute eigenvectors of P_n in the first dimension and eigenvectors of P_m in the second dimension, the resulting eigenvectors will be Kronecker product of both of them, see Subsection 2.4.2.

Again, we are able to compute eigenvalues and eigenvectors $P_n \times P_m$ rewritten from Section 2.4 (as in one–dimension). Consider Laplacian matrix $L(P_n \times P_m)$ of the Cartesian product $P_n \times P_m$ on nm vertices with eigenvalues $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{nm}$ with corresponding eigenvectors $v_{11}, v_{12}, \ldots, v_{nm}$. The eigenvalues are all possible sums of $\lambda_k + \lambda_l$. The eigenvectors are Kronecker products of eigenvectors v_k, v_l.

$$\lambda_{kl} = 4 - 2\cos\left(\frac{\pi(k-1)}{n}\right) - 2\cos\left(\frac{\pi(l-1)}{m}\right), \qquad (4.26)$$

$$\forall k = 1, 2, \ldots, n,\ l = 1, 2, \ldots, m. \qquad (4.27)$$

$$v_{kl}[(i-1)m + j] = \cos\left(\frac{\pi(k-1)}{n}(i - \frac{1}{2})\right) \cos\left(\frac{\pi(l-1)}{m}(j - \frac{1}{2})\right), \qquad (4.28)$$

$$\forall k = 1, 2, \ldots, n,\ l = 1, 2, \ldots, m. \qquad (4.29)$$

The $v_{kl}[(i-1)m + j]$ denotes the $((i-1)m + j)$-th entry of the vector v_{kl} $(i = 1, \ldots, n,\ j = 1, \ldots, m)$.

Notice that in this notation, the q_{i2} corresponds to eigenvector $v_{1,2}$ and the q_{i3} corresponds to eigenvector $v_{2,1}$. The first eigenvector $v_{1,1}$ is a constant vector (as the first eigenvalue is still zero).

Remark 4.5.1 (One–fixing node in two–dimensions).
Given Cartesian product $P_n \times P_m$ in two–dimensions, where both n, m are odd, we will not consider only $q_{i2} = 0$ (because in two–dimensions, there are exactly m such a points distributed along the second dimension), but also $q_{i3} = 0$ (in two–dimensions, there are exactly n such a points distributed along the first dimension). In crossing of both of these conditions, there is placed one vertex - the best choice of one–fixing node.

Again, thanks to computed eigenvectors, it is possible to express the plane (4.25) in parametric form, set up the solution in sense of Lagrange multiplier and express the exact solution - but the equations became more complicated. We let these manipulations for future work. However, we naturally suppose to get

$$\mathcal{J}(a_1, a_2, \ldots, a_{n-1}) = \lambda_2.$$

for given i such that $q_{i2} = 0 \ \wedge q_{i3} = 0$.

□

Chapter 5

Experiments

5.1 Comparison of Two Main Approaches

Let us compare the two main approaches to detection of one–fixing node, the method using the Perron vector of the adjacency matrix (providing the *capital vertex*) and the method using the eigenvectors of the Laplacian matrix (providing the *cross-eigenvector center*). Exact definitions of these methods are described in Section 3.3. Both methods have been compared on a real example. As an example, the mesh of a screw has been given, see Figure 5.1. The mesh has been decomposed into three subdomains using METIS software [47], and one fixing node in each subdomain has been found using mentioned methods. For better comparison, the *best choice of one–fixing node* satisfying Definition 3.2.1 has been also computed and drawn.

As in Subsection 3.3.4, in Figure 5.1 the following symbols appear:

1. The *best choice of one–fixing node* satisfying Definition 3.2.1 is drawn as a circle ◯.
2. The *capital vertex* satisfying Definition 3.3.4 is drawn as a triangle △.
3. The *cross-eigenvector center* satisfying Definition 3.3.6 is drawn as a star ✳.

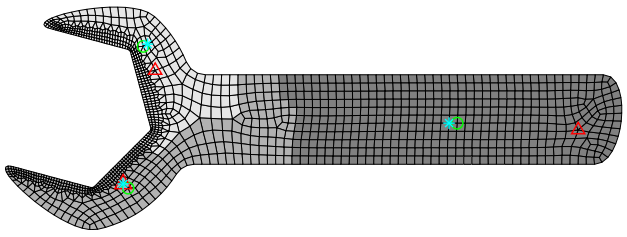


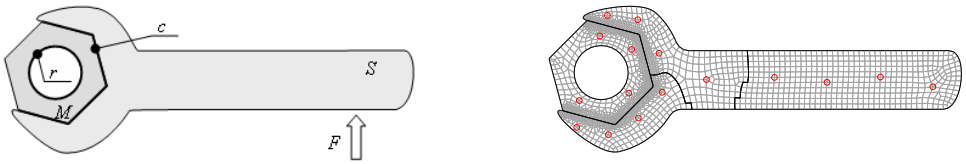
Figure 5.1: Comparison of two approaches

Looking in Figure 5.1, we can see, that the *cross-eigenvector center* provides better approximation of the best choice of one–fixing node than the capital vertex. This is most prominent on the rightmost subdomain, where as the cross-eigenvector center has been detected almost the same vertex as the best choice of one–fixing node, meanwhile the capital vertex has been assigned to the vertex with higher degree which is far from the best choice of one–fixing node. For an exact explanation of this phenomenon see Section 3.3.

5.2 Solution of Contact Problems

To illustrate the practical usage of our fixing node strategy, two contact problems are shown. In both problems, the SPS matrices arise, therefore, the Total-FETI algorithm with fixing node strategy has to be used instead of the standard FETI algorithms.

First, the technical problem depicted in Figure 5.2(a) is presented with corresponding mesh shown in Figure 5.2(b). The mesh is decomposed into four subdomains, each subdomain contains four fixing nodes. The idea of Section 3.2 is used to find fixing nodes in each subdomain; the capital vertex is used as the approximation of fixing node in each subdivision (see Definition 3.3.4).

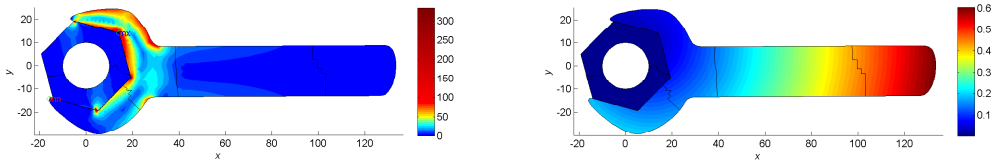


(a) Description of the problem (b) The mesh with 16 fixing nodes

Figure 5.2: The screw & spanner problem

In Figure 5.2(a), there is a mechanical contact between two solid bodies: the screw female M and the spanner S. The screw female M has removed all degrees of freedom on the interior radius r. The spanner S is pressed up by the vertical force F on the right side and the mechanical contact arises along the curve c. Figure 5.3 depicts an is output of Matsol solver [46]. We can see the total stress in Figure 5.3(a) and the total displacement in Figure 5.3(b) respectively.

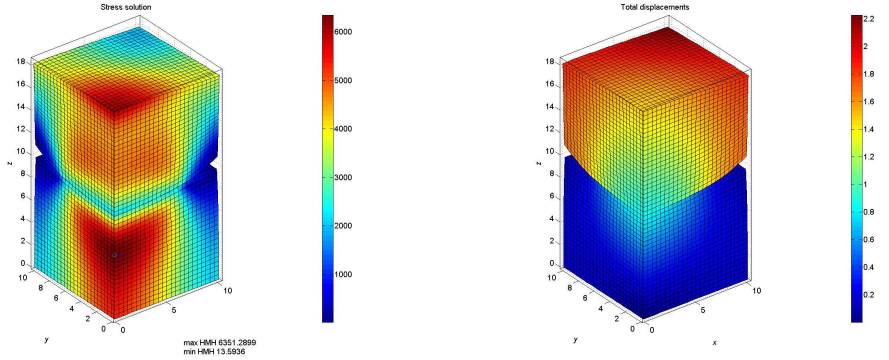
Another practical example is the well known Hertz problem ([35]). The Hertz problem consists of two bodies; the bottom body lies on the base, the



(a) Von Mises stress distribution (b) Total displacements

Figure 5.3: Solution of the screw & spanner problem

upper body is "free in the space", which causes an SPS matrix of the corresponding body. Here, the meshes of each body were decomposed into eight subdomains, each subdomain contained four fixing nodes. The fixing nodes were found in the same way as in the screw & spanner problem. There is the total stress depicted in Figure 5.3(a) and the total displacement depicted in Figure 5.3(b) respectively.



(a) Von Mises stress distribution (b) Total displacements

Figure 5.4: Solution of the Hertz problem

FETI methods are known for their parallel and numerical scalability. The solvability of systems of matrices is measured by their condition numbers. The smaller the condition number is, the faster the algorithms converge.

For such a scalability testing, we have used the mesh from Figure 5.2(b) corresponding to the technical problem depicted in Figure 5.2(a). We have fixed the number of primal variables of discretized problem and we have increased the decomposition parameter, i.e., the number of subdomains from 2 to 32. For each subdomain in each decomposition, four fixing nodes have been chosen. The capital vertex has been used as the approximation of one-fixing node.

In Table 5.1, there are regular condition number cond(K) and cond(K^+), test symmetry of K^+ and three conditions defining the Moore-Penrose generalized inverse. It is evident that the condition number decreases for increasing number of subdomains and that the generalized inverse matrix K^+ is well computed (thanks to the fixing node strategy). In Table 5.1 and also in the graph depicted in Figure 5.5, we can see the decreasing condition numbers of the matrices K and K^+.

No. of domains	2	4	8	16	32
cond(K)	214 840	39 195	11 685	3 374	4 118
cond(K^+)	428 110	40 868	11 921	3 533	4 430
$\frac{\|\|K^+-K^{+'}\|\|}{\|\|K^+\|\|}$	1.1e-015	1.1e-015	1.2e-015	6.0e-016	4.3e-016
$\frac{\|\|K*K^+-(K*K^+)'\|\|}{(\|\|K\|\|*\|\|K^+\|\|)}$	7.3e-005	4.5e-004	9.2e-004	2.9e-003	1.8e-003
$\frac{\|\|K*K^+*K-K\|\|}{\|\|K\|\|}$	6.5e-013	7.4e-014	2.1e-014	6.9e-015	1.1e-014
$\frac{\|\|K^+*K*K^+-K^+\|\|}{\|\|K^+\|\|}$	6.6e-013	2.7e-013	2.0e-014	1.9e-014	1.6e-014

Table 5.1: Condition numbers

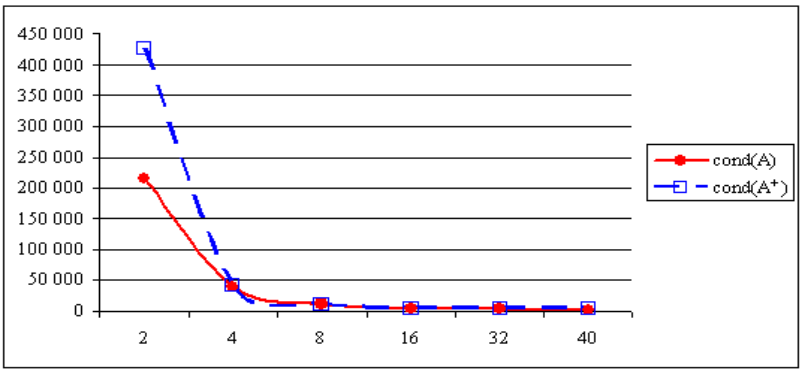


Figure 5.5: Condition number in dependence on the number of subdomains

5.3 Large Scale Problem

The fixing node strategy is used as a part of massively parallel implementation of Total-FETI solver. Several aspect and the other essential steps in efficient implementation of Total-FETI algorithm, as well as a benchmark bellow, can be found in [44].

As a benchmark we have used a 2D elastostatic problem of the steel traverse represented by a domain $\Omega = (0, 600) \times (0, 200)$ with the sizes given in [mm] (see Figure 5.6). The material constants are defined by the Young modulus $E = 2.1 \cdot 10^5$ [MPa], the Poisson ratio $\nu = 0.3$, and the density $\rho = 7.85 \cdot 10^{-9}$ [ton/mm^3]. The body is fixed in all directions along the left side $\overline{\Gamma}_U = \{0\} \times [0, 200]$ and loaded by gravitational forces with the gravity acceleration $g = 9800$mm/s^2.

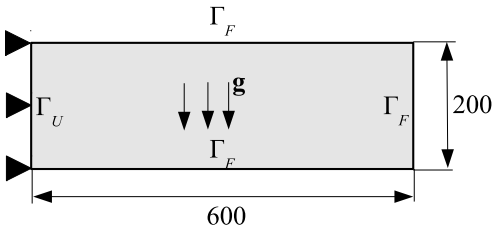


Figure 5.6: Benchmark geometry

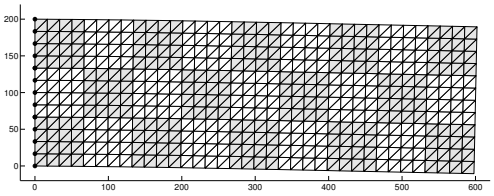


Figure 5.7: Displacements

We varied decomposition and discretization parameters in order to demonstrate the scalability of our method. For these computations we used *HECToR* phase 2b system at EPCC in Edinburgh offering 1856 XE6 compute nodes. Each compute node contains two AMD 2.1 GHz 12-core processors giving a total of 44,544 cores. Theoretical peak performance is around 373 Tflops. There were 32 GB of main memory available per node, which is shared between its 24 cores, the total memory is 58 TB. The processors are connected with a high-bandwidth interconnect using Cray Gemini communication chips. The Gemini chips are arranged on a 3 dimensional torus.

To illustrate the scalability of the Total FETI we decomposed the body Ω into identical square subdomains with the side length H (see Figure 5.7). We gradually chose decompositions into 8×24, 16×48, 24×72, 32×96, and 40×120 boxes. All subdomains were further discretized by the same uniform triangular meshes characterized by the discretization parameter h and the ratio $H/h = 180$. An example of the deformed body together with the traces of decomposition for the choice of parameters $h = 16.7$mm and $H = 66.7$mm is depicted in Figure 5.7.

The Total FETI scalability results are summarized in Table 5.3 and in Figure 5.8.

No. of subdom.	192	768	1728	3072	4800
No. of cores	192	768	1728	3072	4800
Primal variables	12,580,224	50,320,896	114,476,544	201,283,584	314,505,600
Dual variables	129,984	537,216	1,228,464	2,183,424	3,422,400
Kernel dimension	576	2304	5184	9216	14,400
PCGP iterations	42	42	42	42	42
Preproc. time	66.80468	68.03465	71.769	73.57246	78.20962
Solution time	26.7946	27.2203	28.9077	31.7057	40.5153
Total time	93.5992	95.2549	100.6767	105.2782	118.7249
Time per it.	0.5954	0.6041	0.6390	0.6932	0.8757
$\mathbf{F}$ action	0.5791	0.5791	0.5889	0.5771	0.5788
$\mathbf{P_G}$ action (2x)	0.0025	0.0117	0.0382	0.1034	0.2906
$\overline{\mathbf{F}^{-1}}$ action	0.0145	0.0230	0.0487	0.1118	0.2957

Table 5.2: Performance of the Total FETI implementation

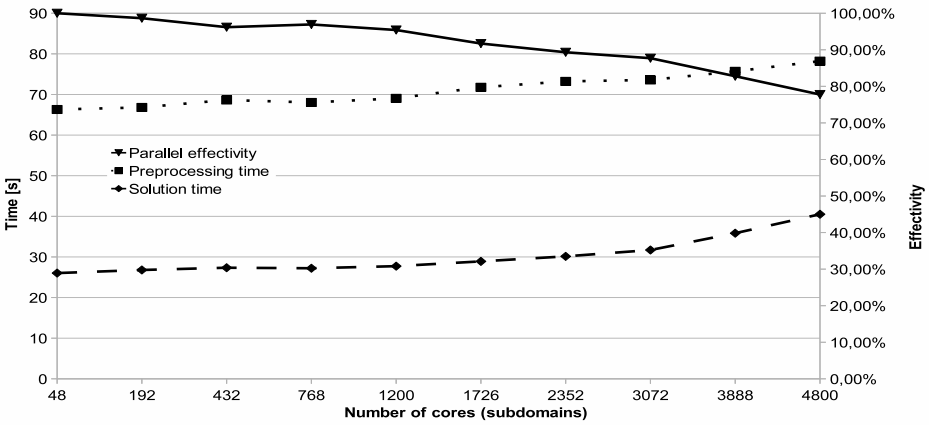


Figure 5.8: Scalability behaviour

We can observe that the number of matrix–vector multiplications stays constant and solution and preprocessing times increase only moderately in agreement with the theory.

Conclusion

In this thesis, I have introduced some improvements of several important parts in the algorithms for FETI (Total FETI respectively) methods together with the theoretical background with main intention to spectral graph theory.

First, I have described a domain decomposition as an essential part of parallel computation of FETI methods. The domain decomposition is closely related to the partitioning of meshes or graphs. At present, there are known several mainstreams methods in partitioning techniques such as a geometric partitioning, hypergraph and graph partitioning, and several best known software packages based on this techniques. Unfortunately, none of these software packages is able to provide an optimal decomposition needed for FETI algorithm. Hence, I have proposed some post-processing on the output of METIS software [47] to compute the optimal decomposition according to our requirements.

The main part of this thesis is dedicated to understand the meshes arising in the discretization of numerical problems from graph theory point of view and to analyze these meshes as graphs. This enables us to use some techniques of the spectral graph theory directly to the meshes of numerical problems. I have introduced a new approach how to use the graph theory techniques in an effective solution of linear systems of equations with SPS matrices. The particular purpose of this approach is to find fixing nodes in the meshes to reduce numerical instability in Cholesky-SVD method, especially when applied to solving the discretized version of Neumann problem (stabilizing action of generalized inverse of SPS matrix).

I have brought the exact definition of fixing node according to its numerical implementation as well as the explanation of the difference between the (one–)fixing node and the best choice of (one–)fixing node. This definition was naturally extended to more fixing nodes. As finding a fixing node according to its definition is not possible because of its numerical complexity, the problem of finding fixing nodes has been translated to the problem of finding graph center. The spectral graph theory techniques have been also shown as a powerful tool for fixing node's detection.

I have provided several candidates to fixing node based on the (spectral) graph techniques together with experimentally results. Using the graph center tends to be only a theoretical tool because of the hight time complexity needed for its computation. One of the candidates more appropriate for numerical computation is the capital vertex that uses the Perron eigenvector of the adjacency matrix. This vector is easy to find (from numerical point of view). Unfortunately, it does not always provide a good approximation of fixing node. I have proposed another candidate: the cross-eigenvector center based on the eigenvectors of the Laplacian matrix of given graph. This candidate approximate the fixing node well even for some special cases of meshes (meshes with non-uniform elements).

One of the valuable assets of this work is the numerical analysis of the choice of cross-eigenvector center as the best choice of one–fixing node. I have reformulated the best choice of one-fixing node into a problem of maximizing of the minimal eigenvalue of a matrix with deleted i-th row and column. In this thesis, I have shown the analysis on one–dimensional problem and I have presented necessary steps to extend these considerations into two–dimensions, which is similar approach but with more complicated expressions. The proper analysis in two–dimensions is left for future work.

The fixing node strategy turns out to be powerful ingredient in FETI based methods where the need to evaluate the action of the generalized inverse arises. This strategy can be simply adapted to non-singular or ill conditioned stiffness matrices. The strategy has been tested on the solution of large contact problems of elasticity. One large scale example (hundred millions of unknowns) is brought in Section 5.3. Another example discretized by more then 10 million variables has been computed, for example, by Dostál, Kozubek, Vondrák in [21]. Their paper deals with scalable TFETI algorithm for the solution of coercive multi-body contact problems of elasticity. In their algorithm, the SPS matrices arise and the generalized inverse of these matrices are solved by choosing a non-precisely defined fixing node.

Further work in this field will consists in discovering suitable technique(s) to find a good approximation of more fixing nodes. One of suitable techniques could be usage of eigenvectors of higher order.

As the eigenvectors corresponding to the several smallest eigenvalues of the Laplacian matrix are harder to compute than the eigenvector corresponding to the largest eigenvalue of the adjacency matrix, the research of the fast algorithm to compute these eigenvectors even for large-scale and bad conditioned problems can be also appropriate topic for further research.

The eigenvector analysis plays a role in different fields of study, thus practical applications of these results to different topics will be also concerned.

Bibliography

[1] Anderson W. N., T. D. Morley: Eigenvalues of the Laplacian of a graph. Linear and Multilinear Algebra, 18:141-145 (1985).

[2] Bapat R. B., S. Pati: Algebraic connectivity and the characteristic set of a graph. Linear Multilinear Algebra, 45(23):247273, (1998).

[3] Barik S., S. Pati: On algebraic connectivity and spectral integral variations of graphs. Linear Algebra Appl., 397:209222, (2005).

[4] Barnard S. T., H. D. Simon: A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems. Concurrency: Practice and Experience, 6(2):101-117, (1992).

[5] Berger M., S. Bokhari: A partitioning strategy for nonuniform problems on multiprocessors. IEEE Trans. Computers, C-36, 570-580 (1987).

[6] Biyikoglu T., J. Leydold, and P. F. Stadler: Laplacian Eigenvectors of Graphs. Lecture Notes in Mathematics, Springer-Verlag Berlin Heidelberg, ISBN 978-3-540-73509-0 (2007).

[7] Bochev P., R. B. Lehoucq: On the Finite Element Solution of the Pure Neumann Problem. SIAM Rev. 47, 50–66, (2005).

[8] Bouchala J., Dostál. Z., and Sadowská, M.: Scalable BETI for variational inequalities. In Domain Methods in Science and Engineering XVII. Lecture Notes in Computational Science and Engineering (LNCSE) 60, 167–174 (2008).

[9] Brzobohatý T., Z. Dostál, T. Kozubek, P. Kovář, and A. Markopoulos: Cholesky–SVD decomposition with fixing nodes to stable computation of a generalized inverse of the stiffness matrix of a floating structure. submited to Int. J. Numer. Meth. Engng (2010).

[10] Catalyurek U., C. Aykanat: Hypergraph-partitioning-based decomposition for parallel sparse matrix vector multiplication. IEEE Trans. Parallel Dist. Systems, 10(7), 673–693 (1999).

[11] Courant R., D. Hilbert: Methods of Mathematical Physics, Vol. 1. Interscience, New York, (1953).

[12] Cvetkovic D. M., M. Doob, and H. Sachs: Spectra of graphs, Theory and application. Deutscher Verlag der Wissenschaften-Academic Press, Berlin-New York (1980).

[13] Chung F. R. K.: Spectral Graph Theory. Reg. Conf. Series in Math, 92:0160-7642 (1994).

[14] Davies E. B., G. M. L. Gladwell, J. Leydold, and P. F. Stadler: Discrete nodal domain theorems. Lin. Algebra Appl., 336:5160, (2001).

[15] Dostál Z.: Optimal Quadratic Programming Algorithms. Springer Optimization and Its Applications, Vol.23 (2009).

[16] Dostál Z. and D. Horák: Scalable FETI with optimal dual penalty for a variational inequality. Numerical Linear Algebra with Applications, 11, 455-472 (2004).

[17] Dostál Z., D. Horák: Theoretically supported scalable FETI for numerical solution of variational inequalities. SIAM Journal on Numerical Analysis 45(2), 500-513 (2007).

[18] Dostál Z., D. Horák, and R. Kučera: Total FETI – an easier implementable variant of the FETI method for numerical solution of elliptic PDE. Communications in Numerical Methods in Engineering, 22, 1155–1162 (2006).

[19] Dostál Z., D. Horák, R. Kučera, V. Vondrák, J. Haslinger, J. Dobiáš, and S. Pták: FETI based algorithms for contact problems: scalability, large displacements and 3D Coulomb friction. Comp. Meth. Appl. Mech. Eng., 194, 395-409 (2005).

[20] Dostál Z., T. Kozubek, A. Markopoulos, and M. Menšík: Cholesky factorization of a positive semidefinite matrix with known kernel. submitted (2009).

[21] Dostál Z., T. Kozubek, and V. Vondrák: Scalable TFETI algorithm for the solution of coercive multibody contact problems of elasticity. submitted (2009).

[22] Drábek P., G. Holubová: Parciální diferenciální rovnice. Západočeská univerzita v Plzni, ISBN 80-7082-766-1 (2001).

[23] Farhat C., M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen: FETI–DP. A dual–prime unified FETI method. I: A faster alternative to the two–level FETI method. International Journal for Numerical Methods in Engineering 50, 1523–1544 (2001).

[24] Farhat C., F.-X. Roux: A method of finite element tearing and interconnecting and its parallel solution algorithm. International Journal for Numerical Methods in Engineering, 32, 1205-1227, (1991).

[25] Farhat C., M. Géradin: On the general solution by a direct method of a large scale singular system of linear equations: application to the analysis of floating structures. International Journal for Numerical Methods in Engineering, 41, 675–696, (1998).

[26] Felipa C.A., K.C. Park: The construction of free–free flexibility matrices for multilevel structural analysis. Computer Methods in Applied Mechanics and Engineering, 191, 2111-2140 (2002).

[27] Fiedler M.: Algebraic connectivity of graphS. Czech. Math. J. 23(98), 298–305 (1973).

[28] Fiedler M.: A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czech. Math. J. 25(100), 619–633 (1975).

[29] Golub G.H., C.F. Van Loan: Matrix Computations. (2nd edn) John Hopkins University Press: Baltimore (1989).

[30] Greenough Ch., R. Fowler: Software for Partitioning Finite Element Meshes. Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, 0X11 0QX, UK (2005).

[31] Hall F., K. Patel, and M. Stewart: Some Eigenvalue Interlacing Results [manuscript, not published yet] (2009-2010).

[32] Haslinger J., I. Hlaváček, J. Lovíšek and J. Nečas: Riešenie variačných nerovností v mechanike, Alfa (1982).

[33] Haslinger J., and R. Kučera: Discretization and numerical realization of contact problems with orthotropic coulomb friction, SIAM J. Scientific Computing, accepted (2009).

[34] Hellmuth M., W. Imrich, W. Klöckl, and P. F. Stadler: Approximate Graph Products, submitted to European Journal of Combinatorics (2008).

[35] Hertz, H.: Über die Berührung fester elastischer Körper. J. reine und angewandte Mathematik, 92, S. 156–171 (1882).

[36] Horák D.: FETI based domain decomposition methods for variational inequalities. Ph.D. Thesis (2007).

[37] Horn R.A., Ch.R. Johnson: Matrix Analysis. Cambridge University Press, ISBN 978-0-521-38632-6 (1985).

[38] Hendrickson B., R. Leland: The Chaco User's Guide Version 2.0. Tech. Rep. SAND95-2344, Sandia Natl. Lab., Albuquerque, NM 87185-1110 (1995).

[39] Justino M.R., K.C. Park, and C.A. Felipa: The construction of free–free flexibility matrices as generalized stiffness matrices. International Journal for Numerical Methods in Engineering 40, 2739–2758 (1997).

[40] Kabelíková P.: Graph partitioning using spectral methods. Institute of Geonics, Ostrava, 978-80-86407-12-8, (2007).

[41] Kabelíková P.: Graph partitioning using spectral methods. Master Thesis (2006).

[42] Kabelíková P.: Graph centers used for stabilization of matrix factorizations, in Discussiones Mathematicae Graph Theory Journal 30(2), p. 249-259, ISSN 1234-3099 (2010).

[43] Kirkland S., M. Neumann, and B. L. Shader: Characteristic vertices of weighted trees via Perron values. Linear Multilinear Algebra, 40(4):311325, (1996).

[44] Kozubek T., V. Vondrak, M. Mensik, D. Horak, Z. Dostal, V. Hapla, P. Kabelikova, and M. Cermak: Total FETI domain decomposition method and its massively parallel implementation, *submitted*.

[45] Markopoulos A.: Škálovatelné metody rozložení oblasti k řešení statických úloh mechaniky (in czech). Ph.D. Thesis (2009).

[46] Kozubek T., A. Markopoulos, T. Brzobohatý, R. Kučera, and V. Vondrák: MatSol – MATLAB efficient solvers for problems in engineering. [online] http://www.am.vsb.cz/matsol, VŠB–Technical University of Ostrava (2008).

[47] Karypis G., V. Kumar: METIS manual Version 4.0. [online] http://glaros.dtc.umn.edu/gkhome/views/metis, University of Minnesota (1998).

[48] Maple [online] http://www.maplesoft.com/, University of Waterloo in Waterloo (2012).

[49] Merris R.: Laplacian matrices of graphs: A survey. Lin. Algebra Appl., 197 (198), 143-176 (1994).

[50] Merris R.: Laplacian graph eigenvectors. Lin. Algebra Appl., 278, 221-236 (1998).

[51] Mitchell W.F.: A Refinement-tree Based Partitioning Method for Dynamic Load Balancing with Adaptively Refined Grids. Journal of Parallel and Distributed Computing, 67(4), 417-429 (2007).

[52] Mohar B.: The Laplacian spectrum of graphs. Preprint Series Dept. Math. University E. K. Ljubljana 26, 261 (1988).

[53] Pan C.T.: On the existence and factorization of rank–revealing LU factorizations. Linear Algebra and Its Applications 316, 199-222 (2000).

[54] Papadrakakis M., Y. Fragakis: An integrated geometric–algebraic method for solving semi-definite problems in structural mechanics. Computer Methods in Applied Mechanics and Engineering 190, 6513–6532 (2001).

[55] Powers D. L.: Graph partitioning by eigenvectors. Lin. Algebra Appl., 101:121133, (1988).

[56] Quenell G.T.: Eigenvalue comparisons in graph theory. Pacific Journal of Mathematics, Vol. 176, No. 2, 443–461 (1996).

[57] Saad Y.: Iterative methods for sparse linear systems (2nd ed.), Society for Industrial and Applied Mathematics, 528 p., ISBN 978-0898715347 (2003).

[58] Pellegrini F.: SCOTCH 3.1 User's Guide. Tech. Rep. 1137-96, Lab. Bordelais de Recherche en Informatique, Universite Bordeaux (1996).

[59] Pothen A., H. D. Simon, and K. Liou: Partitioning Sparse Matrices with Eigenvectors of graphs. SIAM J. Matrix Anal. Appl., 11(3):430-452 (1990).

[60] Preis R., R. Diekmann: The PARTY Partitioning-Library, User Guide - Version 1.1. Tech. Rep. tr-rsfb-96-024, University of Paderborn, Paderborn (1996).

[61] Rao C.R.: A note on generalized inverse of a matrix with applications to problems in mathematical statistics. Journal of the Royal Statistical Society Series B 24, 152–158 (1962).

[62] Taylor V. E., B. Nour-Omid: A Study of the Factorization Fill-in for a Parallel Implementation of the Finite Element Method. Intl. J. Numer. Meths. Engrg., 37, 3809-3823 (1994).

[63] Vondrák V., T. Kozubek, O. Vlach, and Z. Dostál. Parallel solution of contact shape optimization problems based on domain decomposition. in Proc. EngOpt (2008).

[64] Walshaw C.: JOSTLE mesh partitioning software. [online] http://www.gre.ac.uk/jostle/, University of Greenwich (2002).

[65] Wikipedia, the Free Encyclopedia. [online] http://en.wikipedia.org/ (2010).

[66] West D.B.: Introduction to Graph Theory. 2nd. ed., p.588, ISBN 0-13-014400-2 (2001).

[67] Williams R. D.: Performance of dynamic load balancing algorithms for unstructured mesh calculations. Concurrency, Practice, and Experience, 3(5), 457-481 (1991).

[68] Zoltan Version 3.0: Parallel Partitioning, Load Balancing and Data-Management Services. [online] http://www.cs.sandia.gov/Zoltan/Zoltan.html, Sandia National Laboratories (2007).

Author's bibliography

Journal international

1. P. Kovář, A. Silber, P. Kabelíková, and M. Kravčenko: On regular distance magic graphs of odd order (2012), submitted to Journal of Combinatorial Mathematics and Combinatorial Computing.
   ... should appear as J_{neimp} *in Scopus*
2. T. Kozubek, V. Vondrak, M. Mensik, D. Horak, Z. Dostal, V. Hapla, P. Kabelikova, and M. Cermak: Total FETI domain decomposition method and its massively parallel implementation (2011), submitted to International Journals of Advances in Engineering Software.
   ... should appear as J_{imp}
3. Kabelíková P.: Graph centers used for stabilization of matrix factorizations, in Discussiones Mathematicae Graph Theory Journal 30(2), p. 249-259, ISSN 1234-3099 (2010).
   ... J_{neimp} *in Scopus*

Proceedings international

1. Horák D., Kabelíková P., Merta M., and Vondrák V..: The OOSol Scalable Library Based on a Domain Decomposition Method, Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, B.H.V. Topping, P. Iványi (Editors), Civil-Comp Press, Stirlingshire, United Kingdom, ISSN 1759-3433, ISBN 978-1-905088-44-7 (2011).
   ... not indexed yet
2. Vondrak V., Kozubek T., Dostal Z., Kabelikova P., Horak D., and Markopoulos A.: Parallel domain decomposition solvers for contact shape optimization problems, Proceedings of the 7th International Conference on Engineering Computational Technology, Ed.B.H.V.Topping, Civil-Comp Press, Stirlingshire, Scotland, ISBN 978-190508839-3 (2010).
   ... in Scopus
3. Kabelíková P., Kozubek, T., Markopoulos, A., and Brzobohatý, T.: Effective Algorithms for Implementation of FETI-Based Domain Decomposition Methods, Proc. of the Twelf International Conference on Civil, Structural and Enviromental Engineering Computing, Ed. B.H.V.Topping, L.F.Costa Neves, and R.C.Barros, Civil-Comp Press, Stirlingshire, Scotland, ISSN 1759-3433, ISBN 978-1-905088-32-7 (2009).
   ... in Scopus

4. C. Chauvin, S. A. Hirstoaga, P. Kabelíková, A. Rousseau, F. Bernardin, and M. Bossy: Solving the uniform density constraint in a stochastic downscaling model, ESAIM:proceedings, Ed. C. Dobrzynski, P. Frey, Ph. Pebay, ISSN 1270-90024 (2008).
   ... in Zentralblatt Math

Proceedings domestic

1. Kabelíková P., Merta M., and Stachoň M.: OOSol Scalable Library Based on Domain Decomposition Methods, in WOFEX 10 proceedings, Václav Snášel, ISBN 978-80-248-2276-1 (2010).
2. Kabelíková P., Dostál, Z., Kovář, P., and Kozubek, T.: Generalised inverse matrix evaluation using graph theory, in Modelling 2009 proceedings, Institute of Geonics AS CR, Ed. Radim Blaheta, Jiří Starý, ISBN 978-80-86407-66-1 (2009).
3. Brzobohatý, T., Kabelíková P., Kozubek, T., and Markopoulos, A.: Fixing Nodes Method for Stabilization of Generalized Inverse Arising in Total FETI Algorithms, in SNA 09 proceedings, Institute of Geonics AS CR, Ed. Radim Blaheta, Jiří Starý, ISBN 978-80-86407-60-9 (2009).
4. Kabelíková P.: Domain Decomposition Software for FETI Methods, in WOFEX 08 proceedings, Ed. Václav Snášel, ISBN 978-80-248-1807-8 (2008).
5. Kabelíková P.: Spectral Graph Partitioning, in WOFEX 07 proceedings, Ed. Václav Snášel, ISBN 978-80-248-1571-8 (2007).
6. Kabelíková P.: Practical use of Fiedler vector to graph partitioning, in GRAPHS 07 proceedings, Ed. Petr Kovář, ISBN 978-80-248-1445-2 (2007).
7. Kabelíková P.: Graph partitioning using spectral methods, in SNA 07 proceedings, Institute of Geonics AS CR, Ed. Radim Blaheta, Jiří Starý, ISBN 978-80-86407-12-8 (2007).